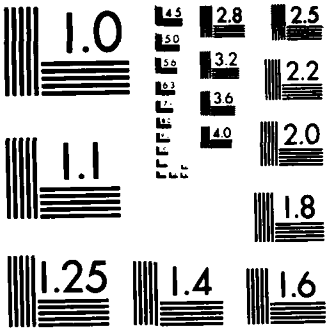
AD-A174 749 EXPERIMENTAL AND THEORETICAL RESPONSE OF MULTIPHASE 1/4
POROUS MEDIA TO DYNAM (U) APPLIED RESEARCH ASSOCIATES
INC SOUTH ROYALTON VT NEW ENGLAND K J KIM ET AL
UNCLASSIFIED 13 AUG 86 ARA-5967-86 AFOSR-TR-86-0665 F/G 8/13 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

EXPERIMENTAL AND THEORETICAL RESPONSE OF MULTIPHASE POROUS MEDIA TO DYNAMIC LOADS

ANNUAL REPORT 1

APPLIED RESEARCH ASSOCIATES, INC.
Engineering and Applied Science

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A since unclassified 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE N/A since unclassified	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 5967-86	5. MONITORING ORGANIZATION REPORT NUMBER(S) **AFOSR-TR- 86-0665**
6a. NAME OF PERFORMING ORGANIZATION Applied Research Assoc. Inc. 6b. OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION N/A
6c. ADDRESS *(City, State, and ZIP Code)* RFD 1, Johnson Hill Road South Royalton, VT 05068	7b. ADDRESS *(City, State, and ZIP Code)* same as 8c
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Air Force Office of Scientific Research 8b. OFFICE SYMBOL *(If applicable)* NA	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F49620-85-C-0102
8c. ADDRESS *(City, State, and ZIP Code)* Building 410 Bolling AFB Washington, DC 20332-6448	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61102F PROJECT NO. 2302 TASK NO. C1 WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)* EXPERIMENTAL AND THEORETICAL RESPONSE OF MULTIPHASE POROUS MEDIA TO DYNAMIC LOADS: ANNUAL REPORT 1

12. PERSONAL AUTHOR(S) Kim, Kwang J.; Blouin, Scott E.; Timian, David A.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Annual	FROM 1 Jul 85 TO 1 Jul 86	13 Aug 1987	300

16. SUPPLEMENTARY NOTATION
N/A

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report summarizes the current status of a combined experimental and theoretical investigation of the response of multiphase porous media to dynamic loadings. This completes the first year of a planned three year investigation. Under the experimental portion, laboratory tests were devised and conducted to measure the compressibility of soil and rock grains containing a large percentage of microporosity. Tests were also developed to model liquefaction due to uniaxial strain loadings and to measure the amount of late-time consolidation as a function of the loading parameters. Finally, a test apparatus to measure fluid friction and energy absorption in porous media under specified flow conditions, including laminar, transient, and turbulent, was designed and constructed. This is currently undergoing evaluation.

Under the theoretical portion of the work, derivations and computational algorithms to model the response of saturated soils and rocks to uniaxial and hydrostatic compressional

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT □ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT □ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Spencer J Wu	22b. TELEPHONE *(Include Area Code)* 767-4963	22c. OFFICE SYMBOL NA

83 APR edition may be used until exhausted.
All other editions are obsolete.

19. Abstract (continued)

loads were developed. The general purpose two-phase code TPDAPII was completely revised to include more realistic plastic and elasto-plastic material models and more efficient computational algorithms. Lastly, theoretical derivations were completed for inclusion in the general purpose multiphase code MPDAP, to be written during the following year's effort.

EXPERIMENTAL AND THEORETICAL RESPONSE OF MULTIPHASE POROUS MEDIA TO DYNAMIC LOADS

ANNUAL REPORT 1

13 August 1986

Prepared for

Air Force Office of Scientific Research
Bolling Air Force Base
Washington, DC 20332-6448

Under Contract No. F49620-85-C-0102

Prepared by

Kwang J. Kim
Scott E. Blouin
David A. Timian

Applied Research Associates, Inc.
New England Division
South Royalton, Vermont 05068

TABLE OF CONTENTS

Section Page

Appendix

SECTION 1

OVERVIEW

1.1 INTRODUCTION

This report summarizes the work performed during the first year of a three year research investigation of the fundamental behavior of multiphase porous materials. The research is organized into two parallel efforts; a laboratory experimental program which is investigating fundamental properties of multiphase porous media and a theoretical program which is formulating advanced multiphase models and incorporating these into numerical algorithms to be used in studying wave propagation phenomena in wet geologies. The experimental program supports the theoretical work by determining fundamental properties and relationships neded in both the modeling and calculational portions of that program. This first annual report summarizes the current status of both efforts. The experimental work is described in Sections 2 through 4 and the theoretical work is summarized in Sections 5 through 7.

1.2 EXPERIMENTAL PROGRAM

Three separate laboratory efforts were initiated during the first year; grain compressibility, shock consolidation, and design and evaluation of a device to be used to measure fluid friction and energy dissipation.

Section 2 describes a series of laboratory tests used to measure the compressibility of various rock and soil grains to peak pressures of about 5 Kbars. A variety of materials were tested, including quartz sand and carbonate soils and rocks. The carbonates were of particular interest because the individual grains contained considerable microporosity which might substantially increase the compressibility of the grains and substantially alter the behavior of the bulk satured material. The carbonates used in these investigations were from nuclear cratering sites at Enewetak Atoll which are currently being studied by the Air Force and the Defense Nuclear Agency.

Section 3 decribes the development and results of laboratory tests to simulate dynamic undrained explosive loadings and subsequent liquefaction and late-time consolidation. These tests, called shock-consolidation tests, are used to evaluate the theoretical modeling of the liquefaction process and to investigate the amount of volume change associated with post-liquefaction consolidation. This post-liquefaction consolidation results in subsidence of the ground surface surrounding an explosion in saturated soil and is thought to be a major cause of the large crater volumes measured on the Pacific nuclear craters. The magnitude of the consolidation/subsidence contribution to those crater volumes is essential to the understanding of cratering mechanics in other geologies.

An important aspect of our three year program is the experimental characterization of fluid friction and energy dissipation in porous media under various flow conditions, including non-steady state and turbulent flow. Section 4 describes the design, fabrication and evaluation of a pressure vessel to be used in the measurement of these parameters during the coming year.

1.3 THEORETICAL PROGRAM

A frequent problem in multiphase modeling is the determination of skeleton properties from undrained test data. Because of the difficulty in conducting these tests and extracting reliable skeleton properties from the test data, it is often preferable to directly measure the ske-leton properties using drained tests. Section 5 describes an analytical procedure for computing the undrained response of fully saturated porous materials in hydrostatic and uniaxial compression using laboratory skeleton properties.

The development of advanced multiphase material models and efficient computational algorithms which incorporate these models is being pursued in two parallel efforts. The first effort, described in Section 6, provides an updated interim means of performing two-phase calculations in support of laboratory and field programs. Advances in material modeling and calculational algorithms are described and incorporated into our advanced two-phase code, TPDAPII.

Section 7 describes the work under the second effort involving the new theoretical formulations to be used in the general multiphase code MPDAP. These new formulations include 4 major advances; incorporation of a fully coupled material model, development of the three-phase model which includes air, a new fluid friction model including both theoretical and empirical formulations, and advances in numerical algorithms which greatly reduce computational times and storage requirements.

SECTION 2

GRAIN COMPRESSIBILITY

2.1 INTRODUCTION

The compressibility of the solid grains is an integral part of the constitutive formulations for multiphase response (e.g. Blouin and Kim, 1984). The compressibilities of the mineral constituents such as quartz and calcium carbonate making up most of the soils of interest, are generally available in the literature from sources such as Bridgman (1931) and Simmons and Wang (1971). During a previous study Blouin et al. (1984) observed that the grains of the carbonate soils from Enewetak Atoll contained a high degree of microporosity within the grains themselves. Figure 2.1 shows a microscopic view of typical beach sand grains in which the micropores, or intragranular porosity, are easily distinguished. The pervasiveness of these micropores explains why the in situ densities of the carbonate soils and rocks are so low. A porosity value of 50% is typical of both the uncemented and cemented sediments at Enewetak. In the case of the beach sand, perhaps a third of the bulk porosity is due to the intragranular microporosity.

If the micro pore space in the carbonate sediments is not fully saturated, resultant undrained compressibilities will be much greater than those predicted using the solid grain compressibilities from the literature. In essence, the bulk moduli of the solid grains would be greatly reduced by the presence of the unsaturated micropores, leading to a much more compressible bulk mixture. Under a high stress load-unload cycle, such a mixture would exhibit permanent compaction and high energy absorption, neither of which are characteristic of fully saturated materials. Thus, it is important to characterize the grain compressibility of materials with microporosity, such as those from Enewetak Atoll, to insure that appropriate relationships for grain compressibility are available and used.

In this section we describe the development of a test technique for characterization of grain compressibility and describe test results on steel and quartz standards as well as on the various materials from Enewetak. Based on these test data it was concluded that the micropores in the Enewetak materials will be very near complete saturation in their in situ state, and will have little influence on the solid grain compressibility.

2.2 TEST PROCEDURES

The technique used to test the samples involved loading an unjacketed soil-fluid mixture hydrostatically to approximately 4 Kbar while insuring that no effective stress develops in the soil skeleton. The total volume change of the mixture is measured and used to compute the stress-strain relationship for the soil utilizing the known pressure volume response of the fluid and the known volume of solid grains. A seven Kbar pressure vessel, with a 1.5 in inside diameter, was used to confine the fluid/grain mixtures. A measured volume of soil is placed in the vessel and saturated with deaired fluid under a vacuum. The fluid/grain mixture is loaded in a servo-controlled press by a piston in the bore of the pressure vessel. The fluid fills the vessel to a level above the solid grains so that the piston never contacts the grains. Thus, only a hydrostatic load is applied to the soil or rock. Figure 2.2 is a schematic section view of the pressure vessel showing the fluid/grain mixture and the piston loading positions.

The first step in measuring the grain compressibility is to accurately define the stress-volume strain response of the pore fluid. In order to accomplish this two tests are performed, one test on the fluid alone, the second test on a fluid-steel mixture, using a steel cylinder of known pressure-volume response submerged in the fluid. Two tests are needed in order to correct the fluid compressibility for the elastic expansion of the pressure vessel. The volume strain of the fluid, ϵ_f, is calculated by using the results from both tests at the same pressure level. For the fluid test, the total volume change may be expressed as

$$\Delta V_{T1} = \epsilon_{f1} V_{f1} + \Delta V_a \qquad (2\text{-}1)$$

and for the fluid-steel mixture test,

$$\Delta V_{T2} = \epsilon_{f2}V_{f2} + \epsilon_s V_s + \Delta V_a \qquad (2\text{-}2)$$

where

ΔV_T = measured total volume change

ϵ_f = volume strain of fluid

ϵ_s = volume strain of steel

V_f = original volume of fluid

V_s = original volume of steel

ΔV_a = apparent excess volume change due elastic expansion of the pressure vessel.

Subtracting Equation 2-2 from Equation 2-1 yields

$$\Delta V_{T1} - \Delta V_{T2} = \epsilon_{f1}V_{f1} - \epsilon_s V_s - \epsilon_{f2}V_{f2} \qquad (2\text{-}3)$$

since ΔV_a should be the same in both tests at any given pressure level. Also since $\epsilon_{f1} = \epsilon_{f2}$, Equation 2-3 can be rewritten as

$$\epsilon_f = \frac{V_{f1} - V_{f2} - \epsilon_s V_s}{(\Delta V_{T1} - \Delta V_{T2})} \qquad (2\text{-}4)$$

which is the fluid volume strain at any particular pressure level. Having thus defined the fluid volume strain, the elastic volume change in pressure vessel, ΔV_a, can be defined as a function of pressure by substitution of Equation 2-4 into Equation 2-1.

Once both the fluid volume strain and vessel expansion relationships are known, the soil grain volume strain can be calculated by

$$\epsilon_g = \frac{\Delta V_T - \epsilon_f V_f - \Delta V_a}{V_g} \qquad (2\text{-}5)$$

where ΔV_T, V_f and V_g are measured and ϵ_f and ΔV_a are the appropriate values from the stress-fluid strain and stress-apparent volume change relationships described above. Equation 2-5 is derived by substituting the strain in the solid grains, ϵ_g, and grain volume, V_g, for the values for steel given in Equation 2-2.

2.3 PRESENTATION OF TEST DATA

Test data, incrementally corrected using Equation 2-5, are presented as plots of pressure versus the volume strain in the solid grains. The slope of the pressure-volume plot represents the tangent bulk modulus of the solid grains. Steel ball bearings and quartz sand were tested in order to verify the technique through comparisons to published values. Selected materials from Enewetak Atoll were also tested including sand and cemented and uncemented cored samples. One sample of cored material was ground to a fine powder and tested to determine whether this mechanical breakdown of the grains had any effect on the influence of microporosity. Figures 2.3 through 2.9 are plots of pressure vs solid grain volume strain for all of the materials tested.

2.4 DATA ANALYSIS

Least-square linear fits to the preliminary data are included in the figures. As noted on the figures, the bulk moduli of the steel ball bearings and the quartz sand compare very well with published data for those materials. The bulk moduli of the Enewetak materials are in the 5 to 6 million psi range except for the beach sand, which has a modulus of 9.1 million psi. The bulk modulus values are summarized in Table 2.1.

Measurement resolution during the initial portion of the loading, below 20,000 psi, was insufficient to determine reliable volume strains. Data below this pressure level are not shown on the data plots. Above 50,000 psi a nonlinear trend is noticed in all of the data which is evidently due to an irregularity in the pressure volume relationship for the pore-fluid. Since the same experimentally determined fluid pressure-volume relationship is used to correct all test data, each of the plots is similarly affected. We are currently running additional fluid compressibility tests to provide better experimental resolution in both the high and low pressure regimes.

2.5 CONCLUSIONS AND RESEARCH PLANS

Our initial study of grain compressibility has provided very promising results, as shown by comparisons to published values. The testing and data reduction procedures are relatively straightforward which makes the test desirable for determining the bulk modulus of unpublished materials. Further work will be completed to resolve the problems encountered in the low and high pressure regimes. Transducers with finer resolution will help define the low pressure behavior, and more testing will be done to work out the irregularities in the high pressure regime.

Table 2.1. Bulk modulus values obtained from grain compressibility testing.

Material	Bulk Modulus	
	(psi x 10^6)	(MPa x 10^4)
Steel Ball Bearings	22.2	15.31
Quartz Sand	5.2	3.58
Enewetak Beach Sand	9.1	6.27
Silt-Sand-Gravel from KAM-2	5.6	3.86
Ground Silt-Sand-Gravel from KAM-2	6.1	4.21
Vugular Limestone	5.7	3.93
Cemented Material from XSA-2	6.1	4.21

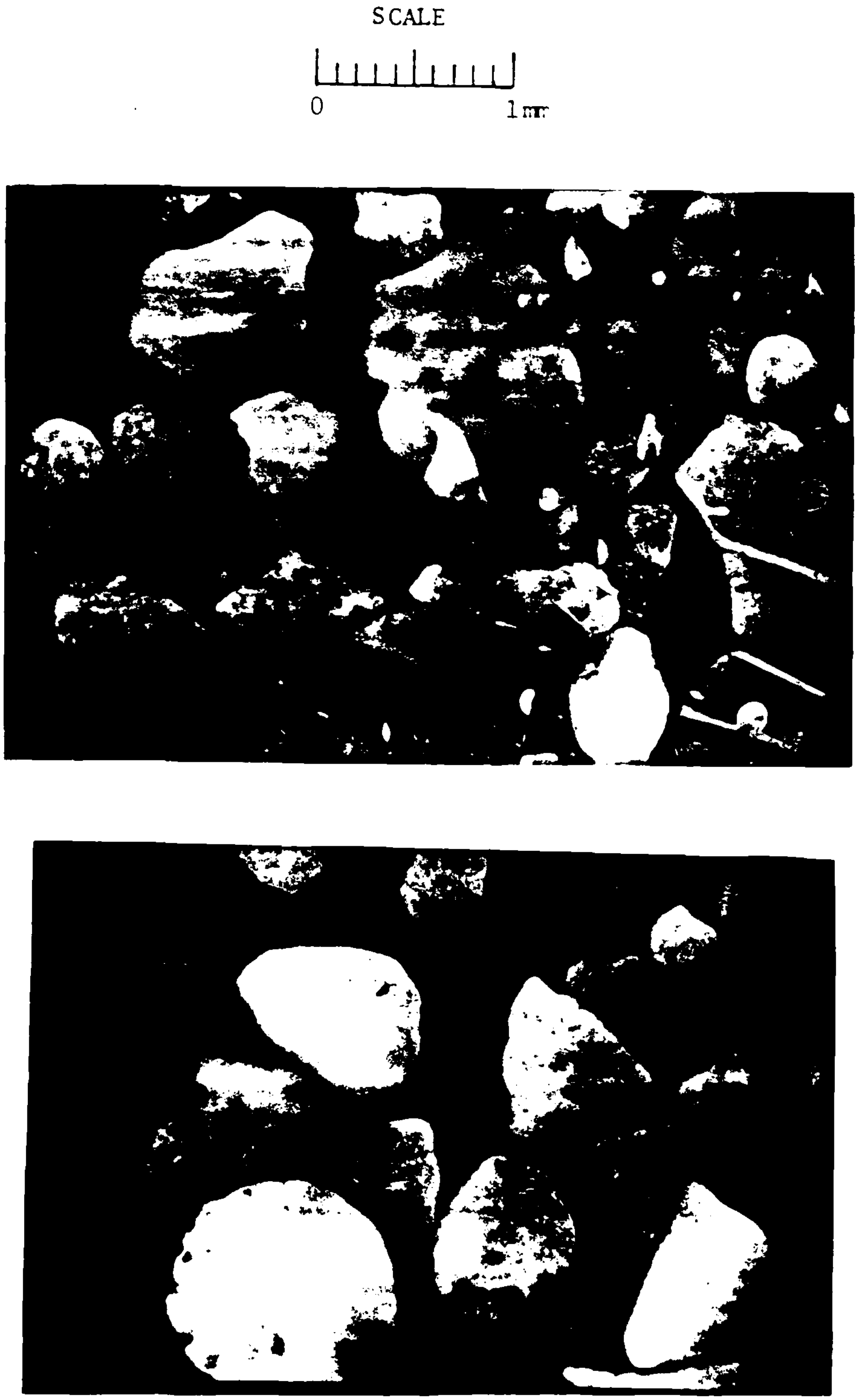


Figure 2.1. Microscopic view of virgin Enewetak beach sand.

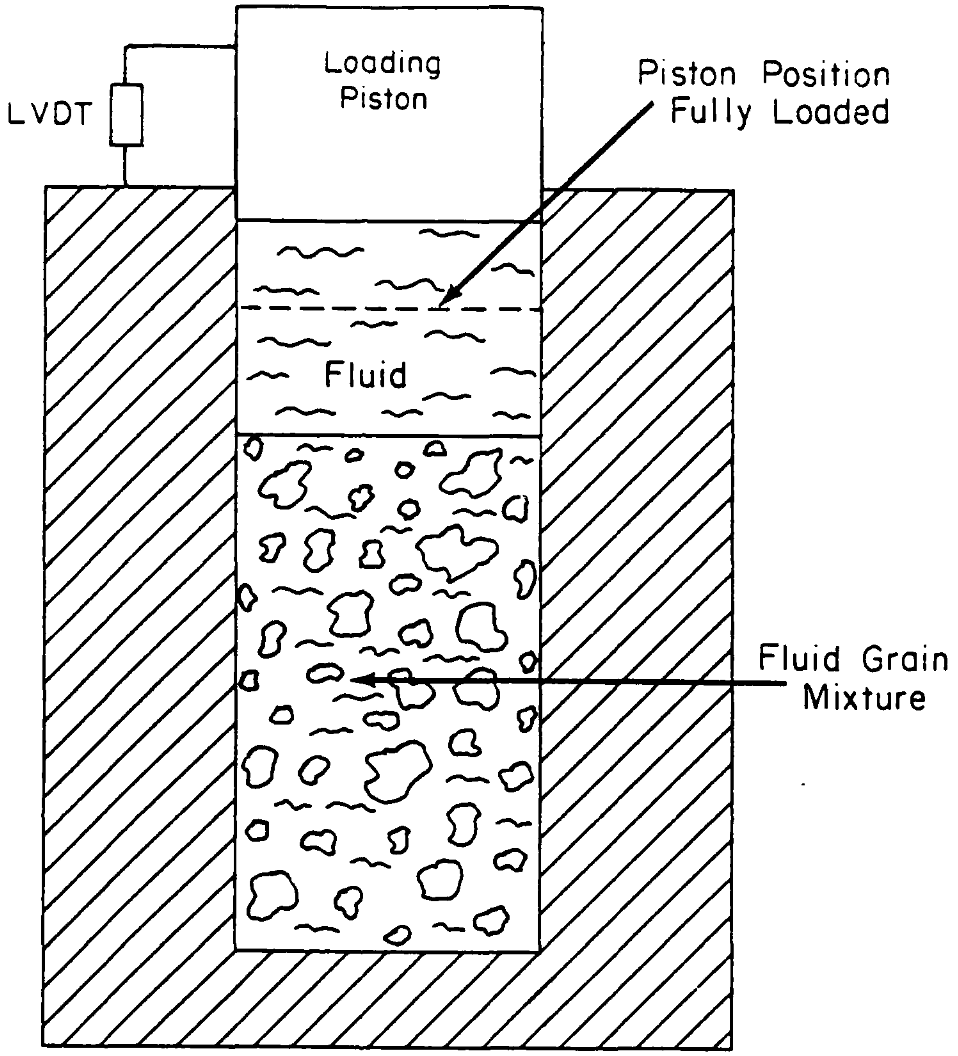


Figure 2.2. Schematic section view of grain compressibility apparatus.

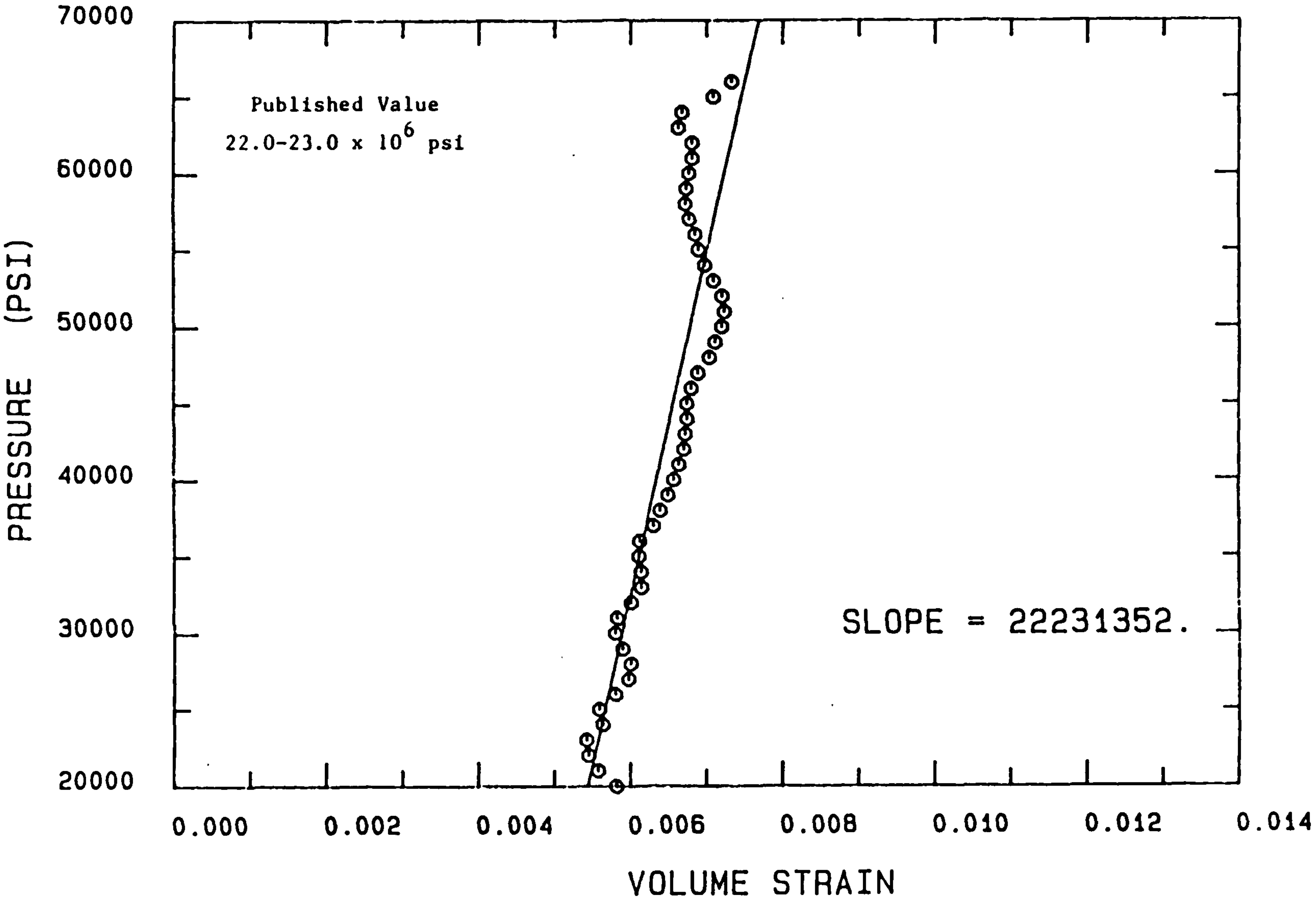


Figure 2.3. Grain compressibility, steel ball bearings.

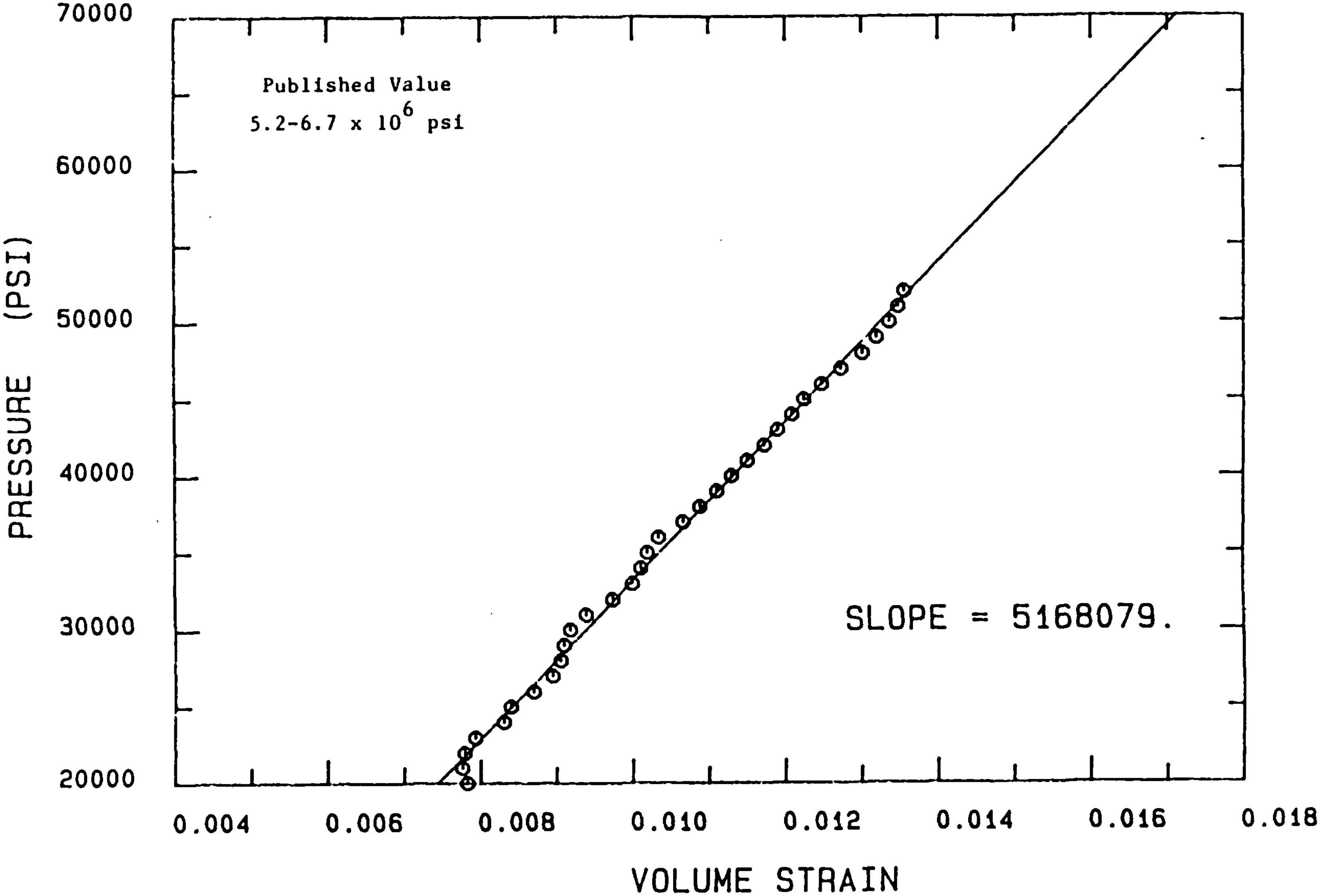


Figure 2.4. Grain compressibility, quartz sand.

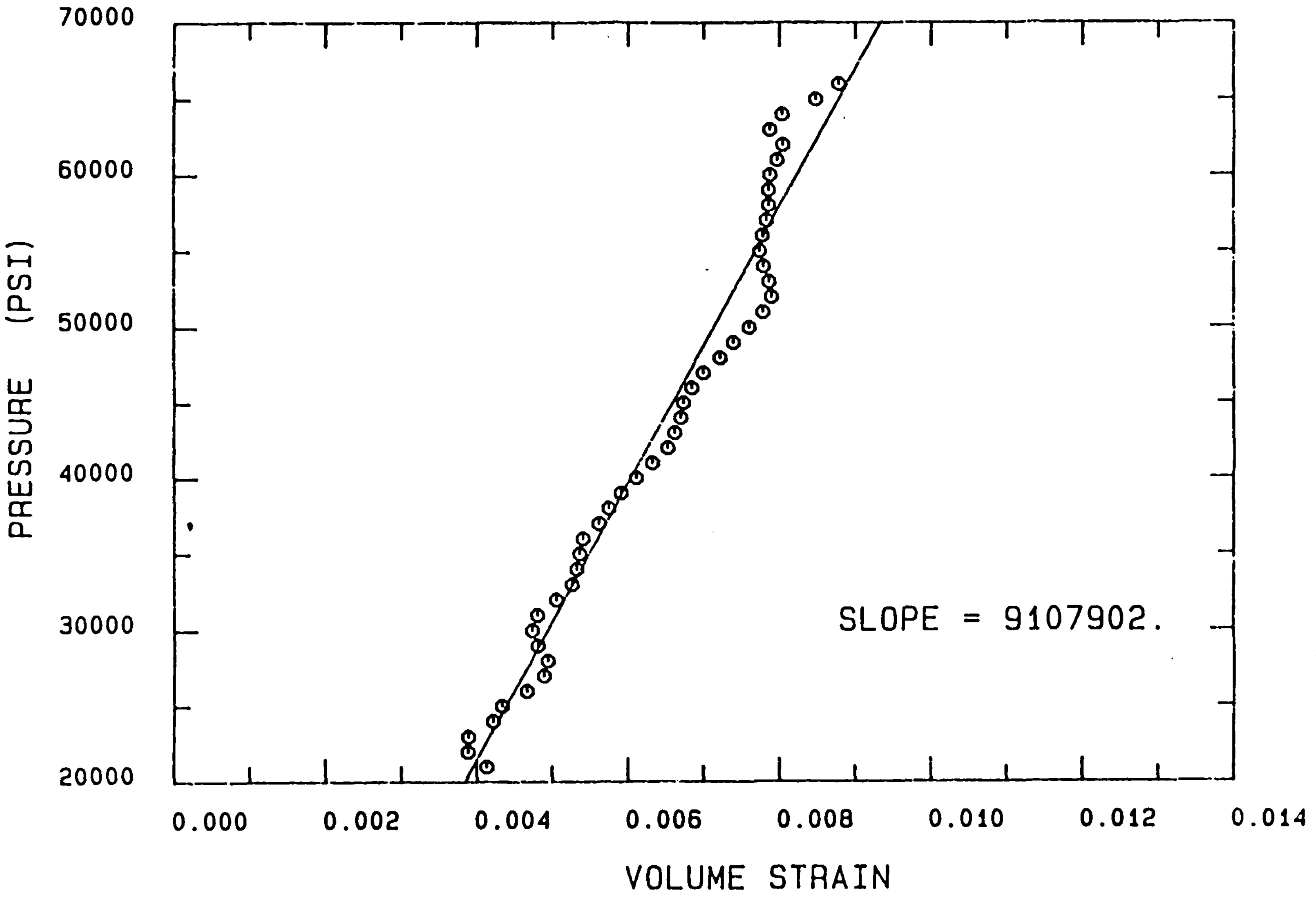


Figure 2.5. Grain compressibility, Enewetak beach sand.

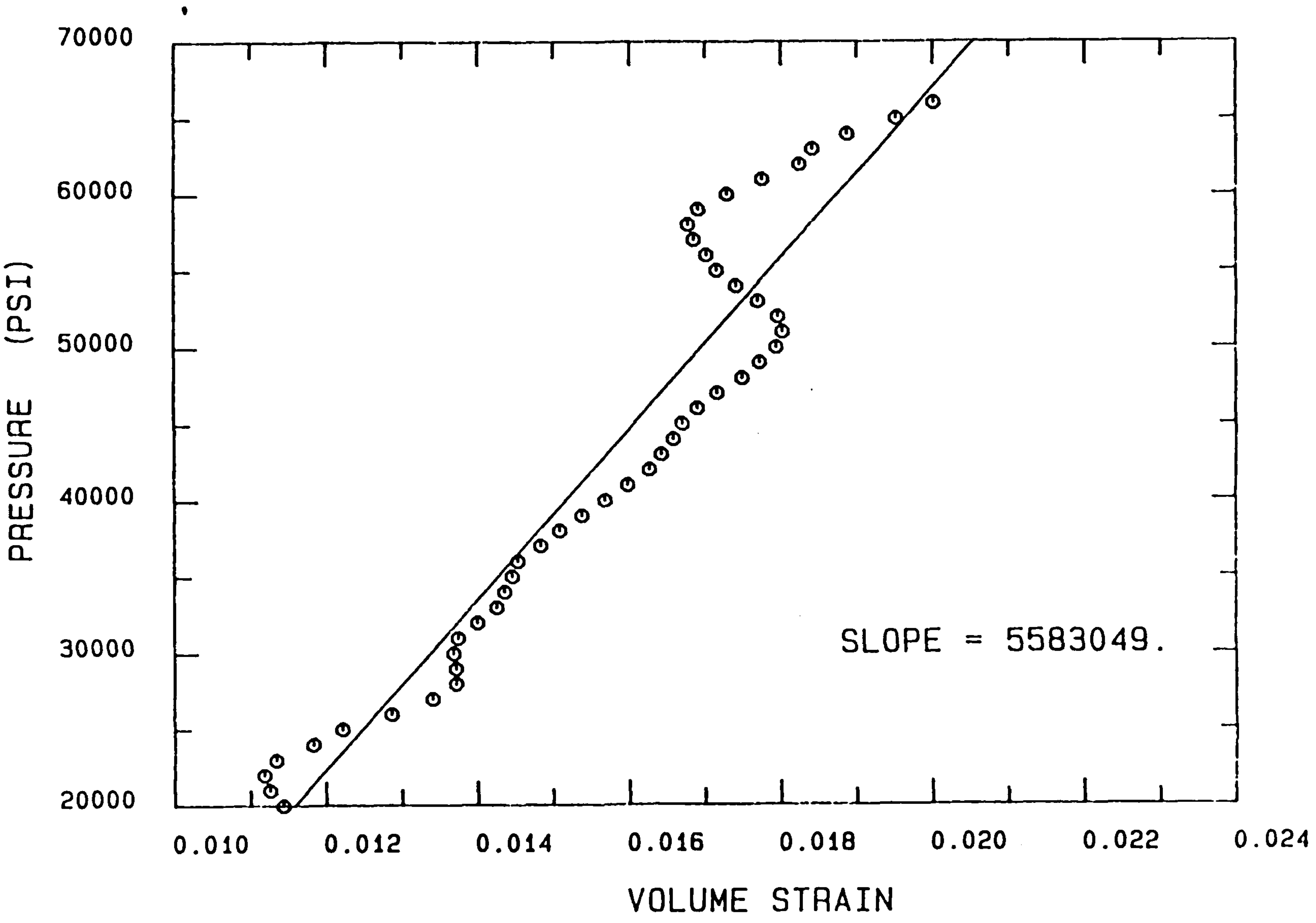


Figure 2.6. Grain compressibility, KAM-2 silt-sand-gravel.

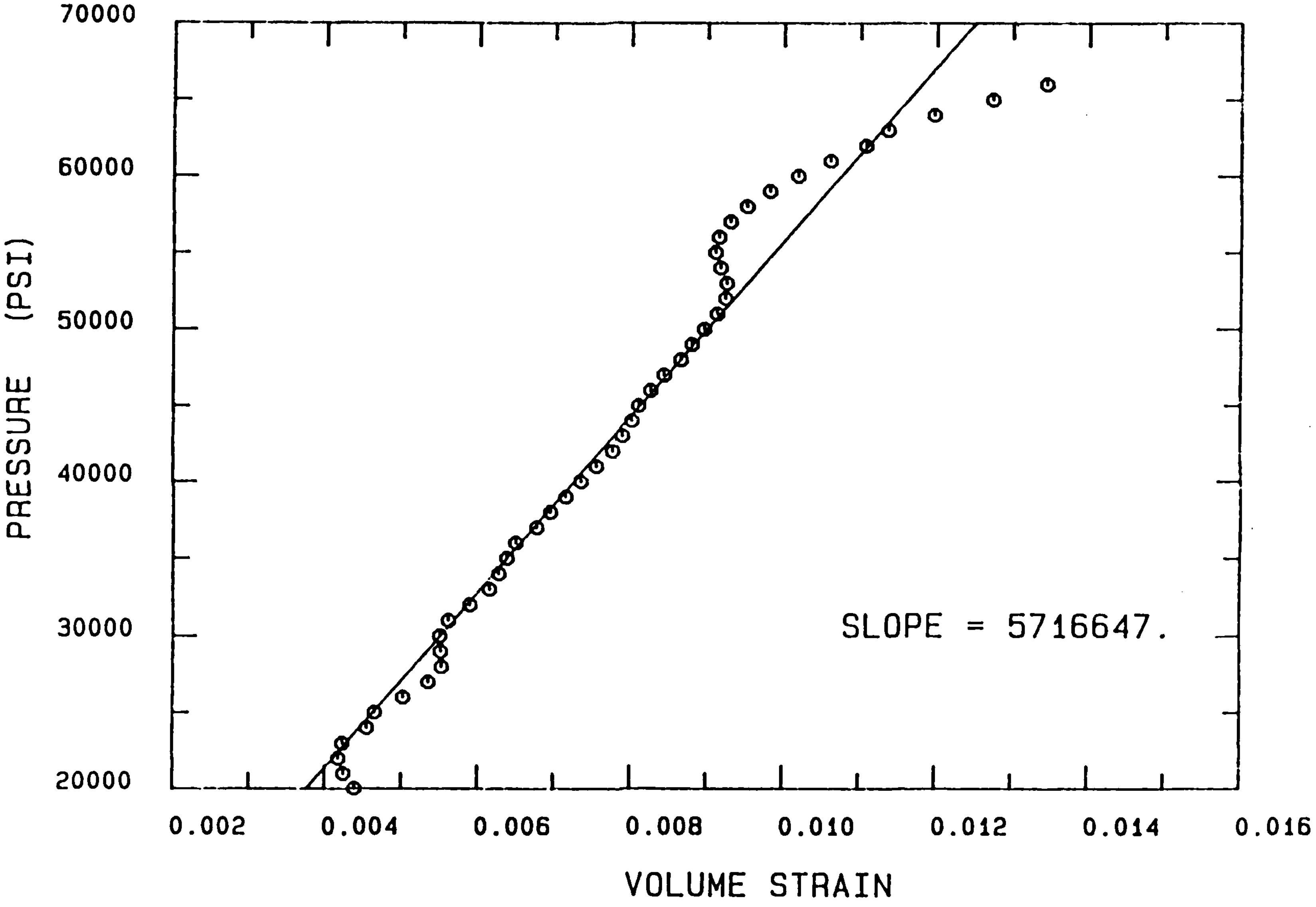


Figure 2.7. Grain compressibility, vugular limestone from KAM-2A.

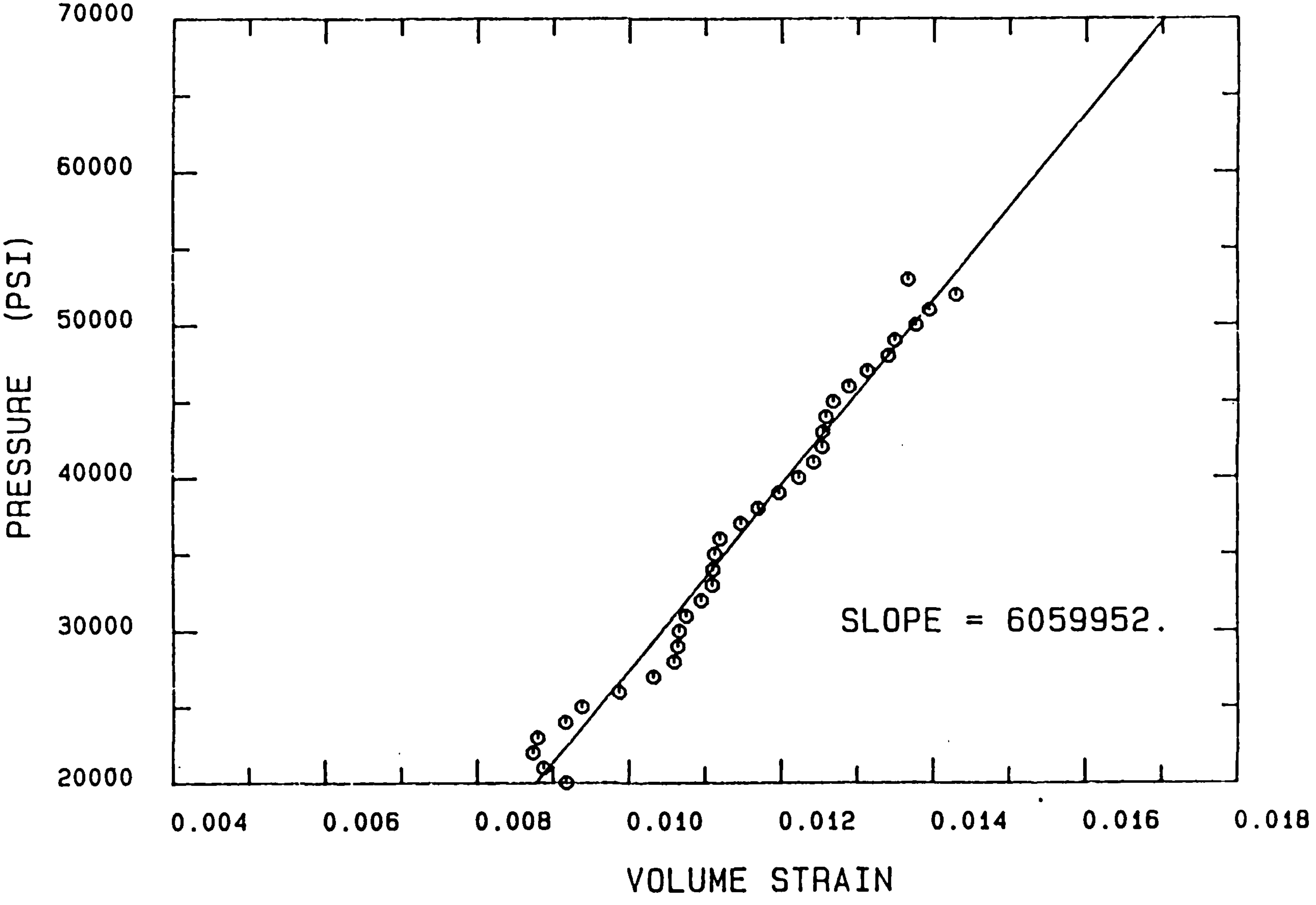


Figure 2.8. Grain compressibility, cemented material from XSA-2.

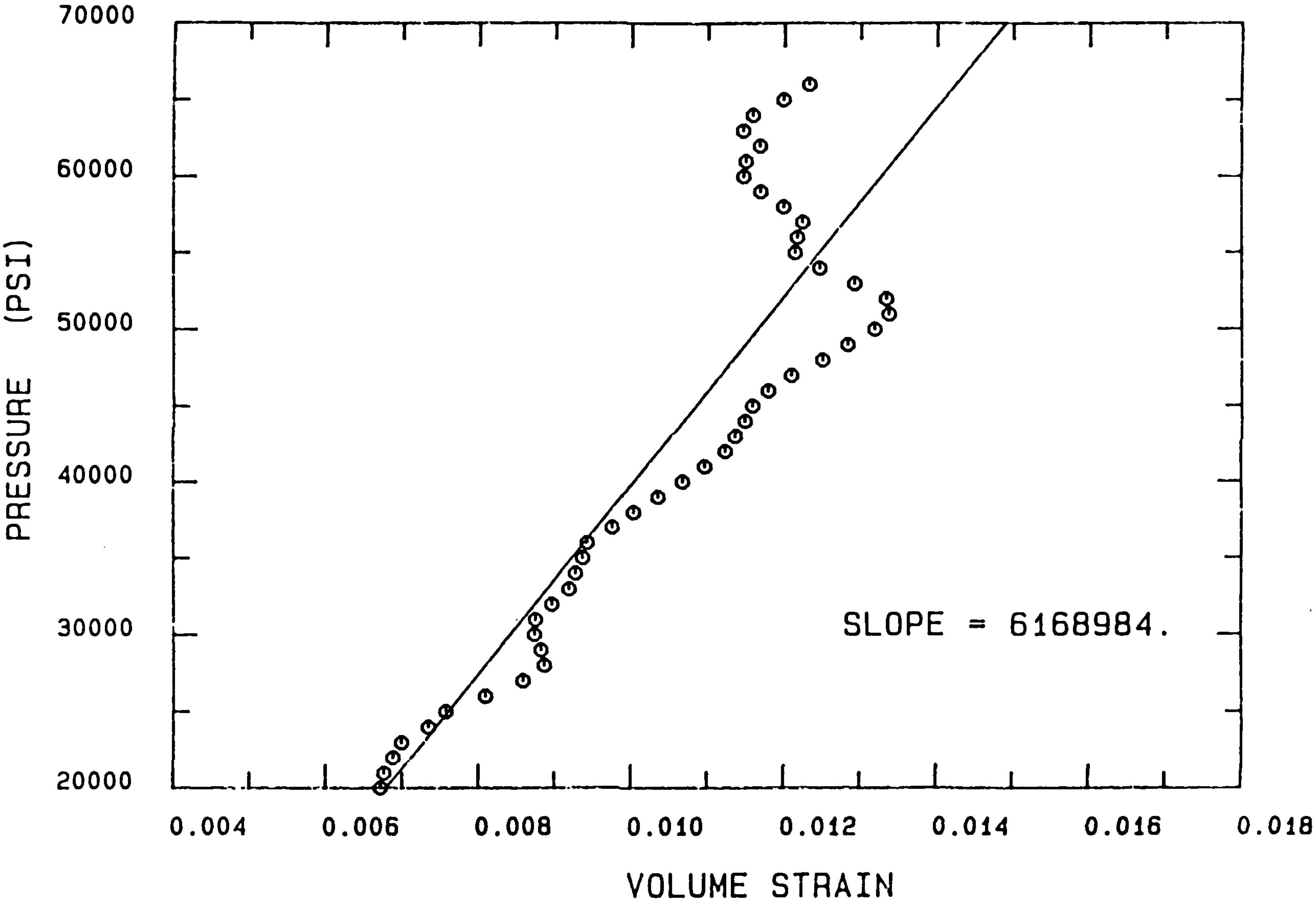


Figure 2.9. Grain compressibility, ground silt-sand-gravel from KAM-2.

SECTION 3

UNIAXIAL STRAIN SHOCK CONSOLIDATION

3.1 INTRODUCTION

In previous work for AFOSR, Blouin and Shinn (1983) provide strong evidence that blast-induced liquefaction and subsequent consolidation of the liquefied deposits can add substantially to the apparent volume of craters in saturated media. In fact, in geologic settings such as Bikini and Enewetak Atolls, consolidation or subsidence may be the dominant mechanism contributing to the crater volumes. While current multiphase models are capable of duplicating the blast-induced liquefaction portion of the crater formation process, no data exist on the subsequent consolidation process to guide the development of more comprehensive theoretical and computational models with which to treat these late-time phenomena.

To meet the need for such phenomenological and quantitative guidance, we have undertaken development of laboratory tests which replicate the important features of the dynamic and quasi-static phases of the shock load-unload and consolidation processes. These tests, termed shock consolidation tests, consist of an undrained load-unload cycle, followed by a drained consolidation loading. The undrained portion is meant to replicate the shock load-unload liquefaction cycle, and the drained portion replicates the late-time consolidation process as in situ effective stresses are reestablished.

This section describes the two versions of the shock consolidation test currently under development and presents preliminary data on both cemented and uncemented porous materials from Enewetak Atoll. Even in the early stages of development, these tests clearly demonstrate the liquefaction process and provide convincing evidence that substantial volume reductions occur during consolidation of the liquefied deposits.

3.2 TEST PROCEDURES

Two types of uniaxial strain snock consolidation tests were developed, oedometer and K_o triaxial. The principal differences between the two test types were in the sample dimensions and in the method of radial restraint which was used. The uniaxial test was selected because it closely approximates the one-dimensional airblast loading and also the uniaxial consolidation condition which is thought to be predominant following liquefaction. Other load-unload strain paths will be utilized in further test development. The oedometer samples were in the form of thin discs (height to diameter ratio of about 0.25) and restrained within a rigid steel ring. The K_o triaxial samples were longer (height to diameter ratio of about 2.0) and were confined by a fluid within a triaxial pressure vessel, such that a zero radial strain condition was maintained during the loading. Other than these basic differences, the testing procedures are similar as described below.

After placing the sample in the test vessel, a vacuum is applied to remove any air in the pores. The sample is then pressure saturated with deaired water. Separate axial force and pore pressure controls allow the sample to be loaded with prescribed initial total stress and pore water pressures. The initial conditions are generally set to reproduce the state-of-stress the soil would experience in situ. A typical set of initial conditions might include an axial effective stress of 200 psi and pore water pressure of 200 psi for a total initial axial stress of 400 psi.

Following the application of the initial stress and pore pressure conditions, the pore water drainage port is closed and an undrained uniaxial loading and unloading cycle is applied. The sample is unloaded back to the original total axial stress. Early in the unloading of both the uncemented and cemented materials, the samples undergo a rapid loss of all effective stress and become liquefied. From this point on the unloading is completely hydrostatic (i.e. total stress equals the pore water pressure).

Once the initial total axial stress is reached, the pore pressure line is opened and pore water is allowed to flow out of the sample under the applied total stress until the initial overburden effective stress is reestablished. During this phase the pore water pressure decreases from the initial total stress value to the initial pore pressure value, while the effective stress increases from zero to its initial value. The total stress is held constant over this time interval.

3.3 PRESENTATION OF TEST DATA

The most important aspect of any consolidation test is the amount of axial deformation. The deformation of interest in the consolidation portion of the shock consolidation test is the change in sample length from the beginning of the drained loading segment to the end of the test. This deformation represents the amount of axial or volume strain which can be expected after the soil undergoes a uniaxial undrained load-unload cycle. Figures 3.1 through 3.3 show preliminary results of an oedometer test on an uncemented silt-sand-gravel sample from Enewetak borehole OAR-2, including plots of total axial stress, pore pressure, and effective axial stress as functions of axial strain. Figures 3.4 through 3.6 show preliminary results for a cemented Enewetak coral. Figures 3.7 through 3.9 show preliminary results of a K_0 triaxial shock consolidation test on saturated beach sand.

3.4 ANALYSIS OF SHOCK CONSOLIDATION DATA

Before beginning analysis of the test data, the important points of the test should be defined. Each plot has been labeled with letters from a to e which represent, respectively, (a) the start of the undrained loading from the initial stress and pore pressure conditions, (b) the peak undrained stress, (c) the point where the effective stress drops to zero and the unloading becomes hydrostatic (liquefaction occurs), (d) the start of the drained consolidation portion of the test, and (e) the point where the final effective stress is once again equal to the initial effective stress. The dashed lines on each of the plots represent the initial stress and pressure values.

Analysis of Figures 3.1 through 3.3 show that the final amount of consolidation for the uncemented material under drained conditions is approximately equal to the deformation which occurred during the undrained portion of the loading. Figures 3.4 through 3.6 show that the final amount of consolidation in the cemented material is slightly less than the peak deformation under the undrained loading. If this proves to be a consistent difference between uncemented and cemented materials, it will likely be due to the larger grain size of the cemented particles after the cementation is broken. The larger average particle sizes will tend to consolidate into a less dense packing under given axial effective stress than will the smaller particles in the uncemented materials. Therefore, the originally cemented samples with their larger particle sizes would be expected to reestablish their original effective stress at a somewhat smaller drained displacement.

A more detailed analysis is currently being performed to calculate drained and undrained modulus values from the shock consolidation tests in the oedometer. Initial work has shown that the cemented moduli are too soft due to excess volume expansion in the pressure vessel. Techniques for correcting for the excess volume expansion similar to those used in the grain compressibility tests are being evaluated.

The K_O triaxial test data presented in Figures 3.7 and 3.8 show behavior similar to the oedometer test results. One basic problem with the K_O triaxial test is that once liquefaction occurs and the unloading becomes hydrostatic, the radial strain cannot be controlled. As a result, the sample deformations during unloading are no longer uniaxial. This difference accounts for the variations in axial and radial deformation observed during the unloading. Generally the samples undergo net radial compression and axial extension, probably due to compression by the sample jacket during the hydrostatic unloading.

3.5 CONCLUSIONS AND RESEARCH PLANS

The initial work completed on the shock consolidation tests indicates that both uncemented and cemented materials undergo liquefaction and subsequent consolidation following undrained uniaxial loadings. Test data also indicate that the volume change due to consolidation may be related to the peak deformation during shock loading. Cementation may reduce the amount of consolidation slightly. More analysis of the test data and additional tests over a broader range of peak stresses are required to determine the influence of the various material and test parameters on the final amount of consolidation.

Additional work is planned in calculating the modulus values for the two phase mixture and the soil skeleton during the undrained loading portion of the test. Also plans have been made to adapt the K_o triaxial shock consolidation test to include a variety of lateral restraint conditions during unloading which might better replicate the strain paths from spherical dynamic loadings. The undrained unloading phase will allow controlled radial expansion. The addition of this expansion may increase the amount of axial deformation that is measured during the drained consolidation portion of the test.

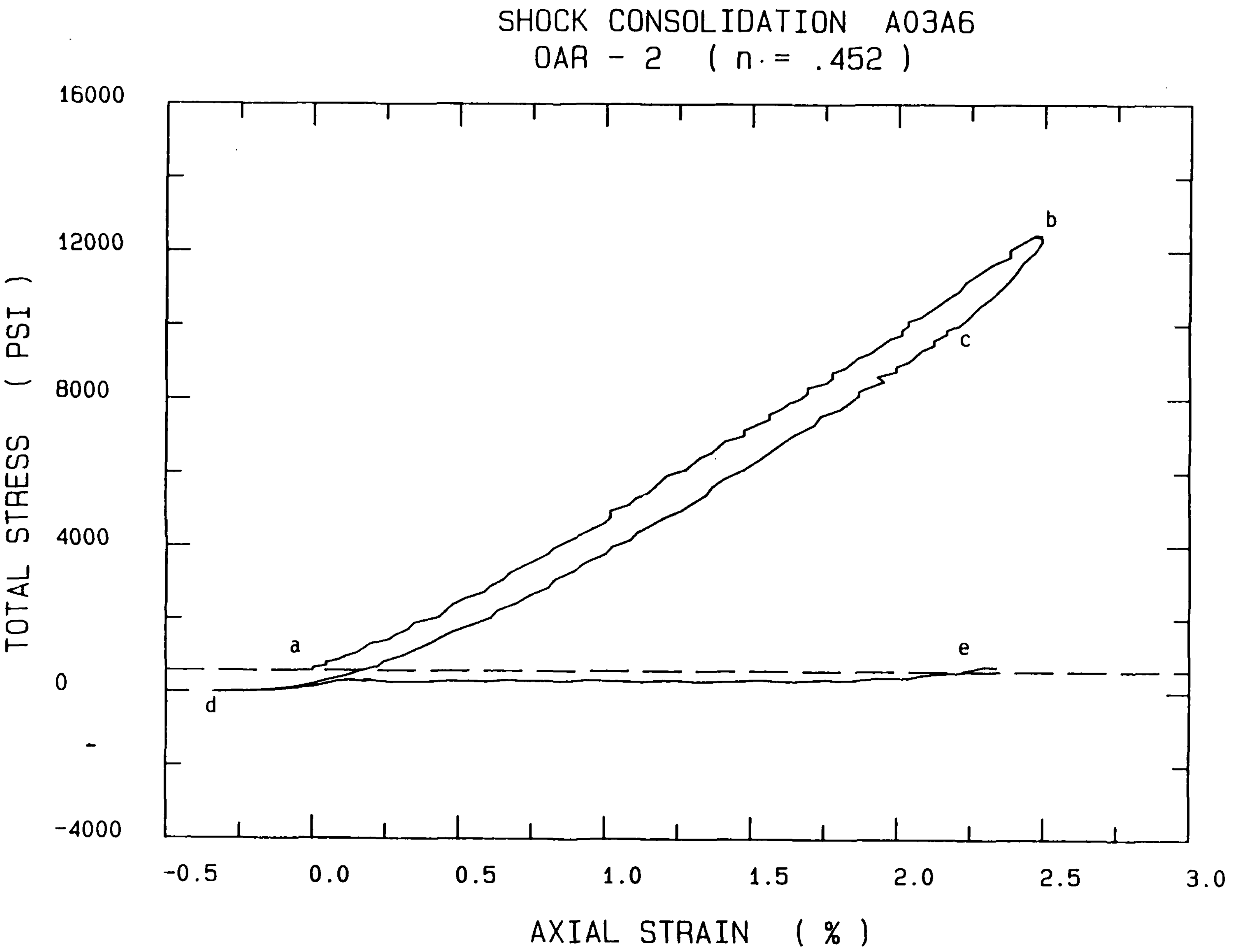


Figure 3.1. Total stress vs. axial strain for a silt, sand, gravel sample.

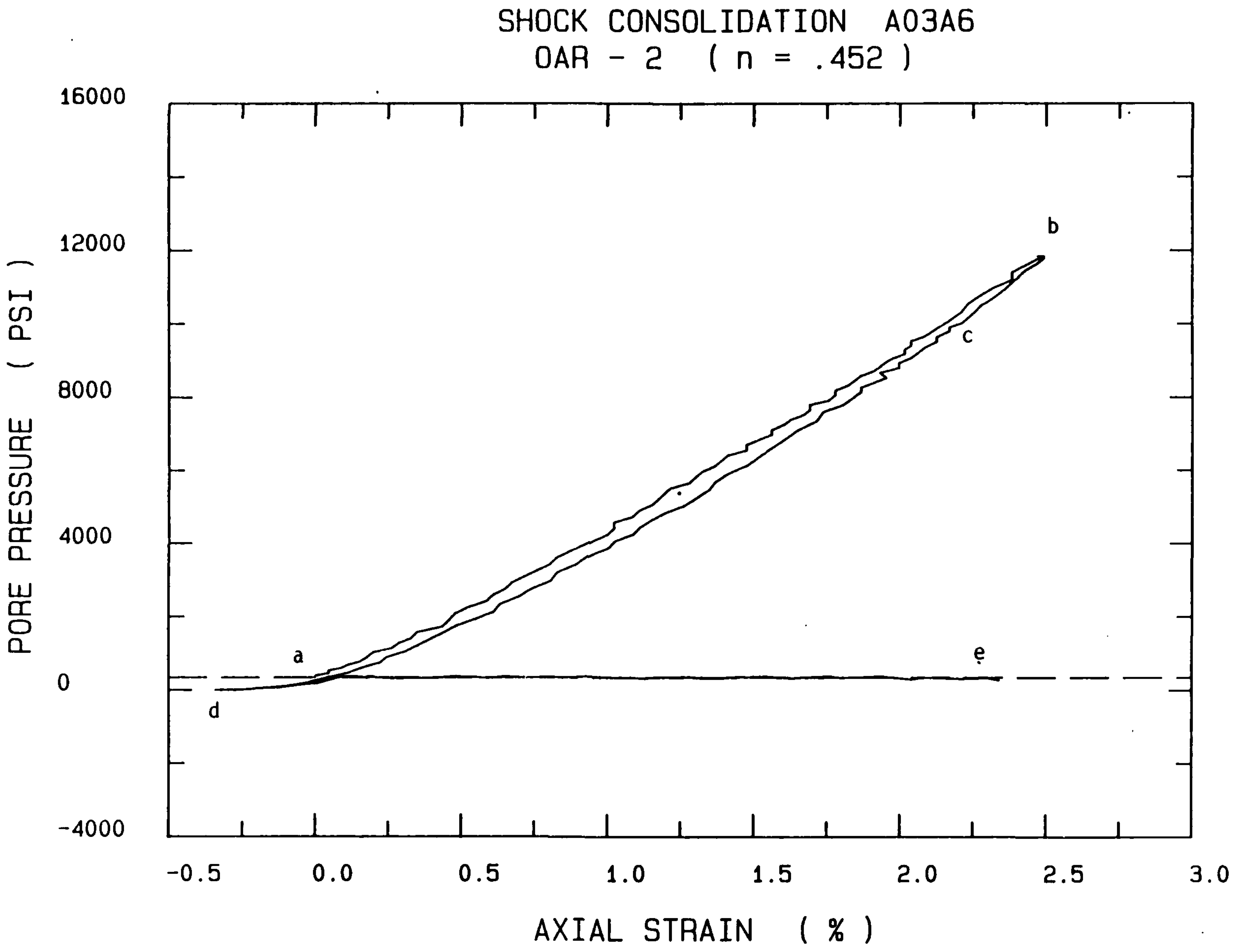


Figure 3.2. Pore pressure vs. axial strain for a sand, silt, gravel sample.

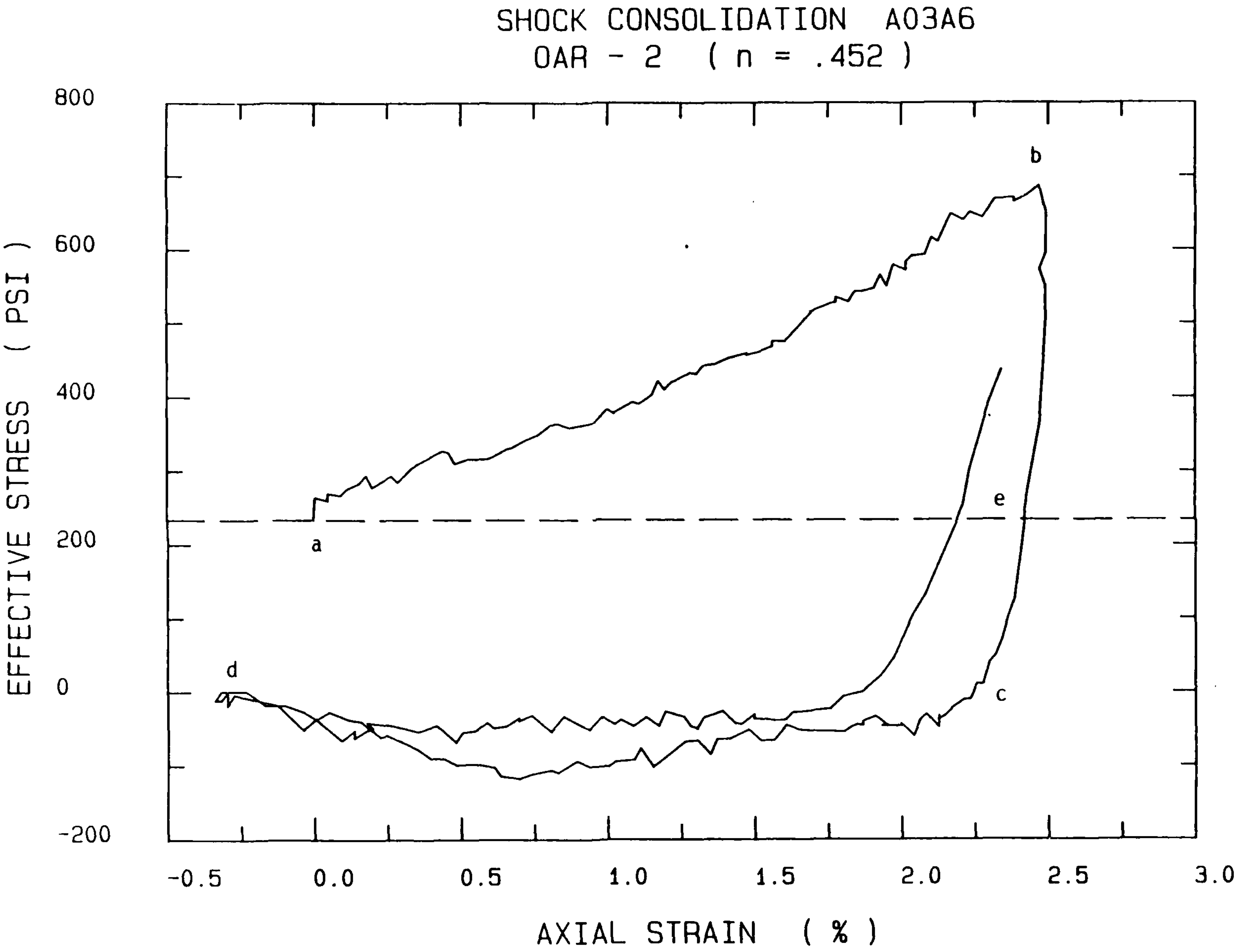


Figure 3.3. Effective stress vs. axial strain for a silt, sand, gravel sample.

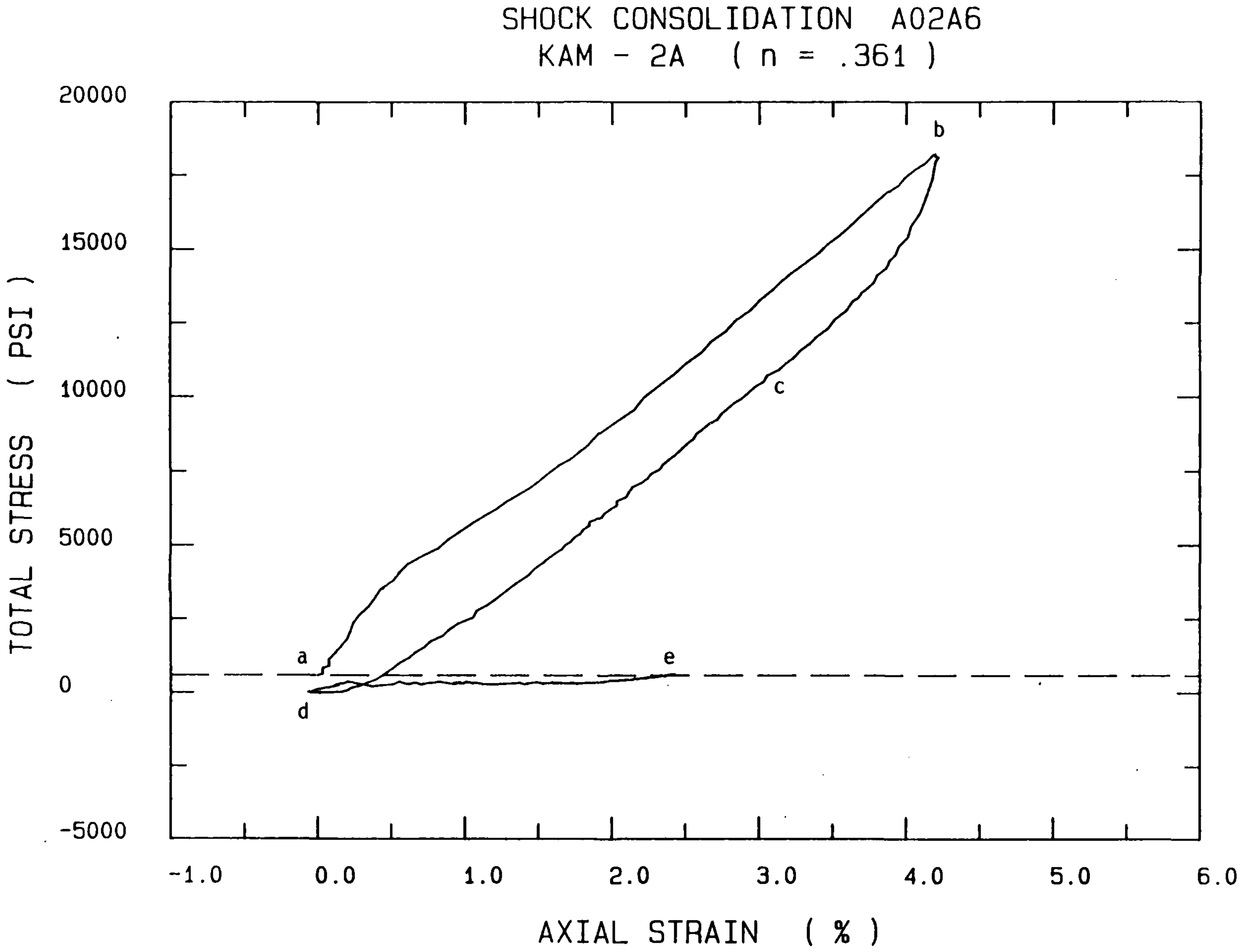


Figure 3.4. Total stress vs. axial strain for a cemented limestone sample.

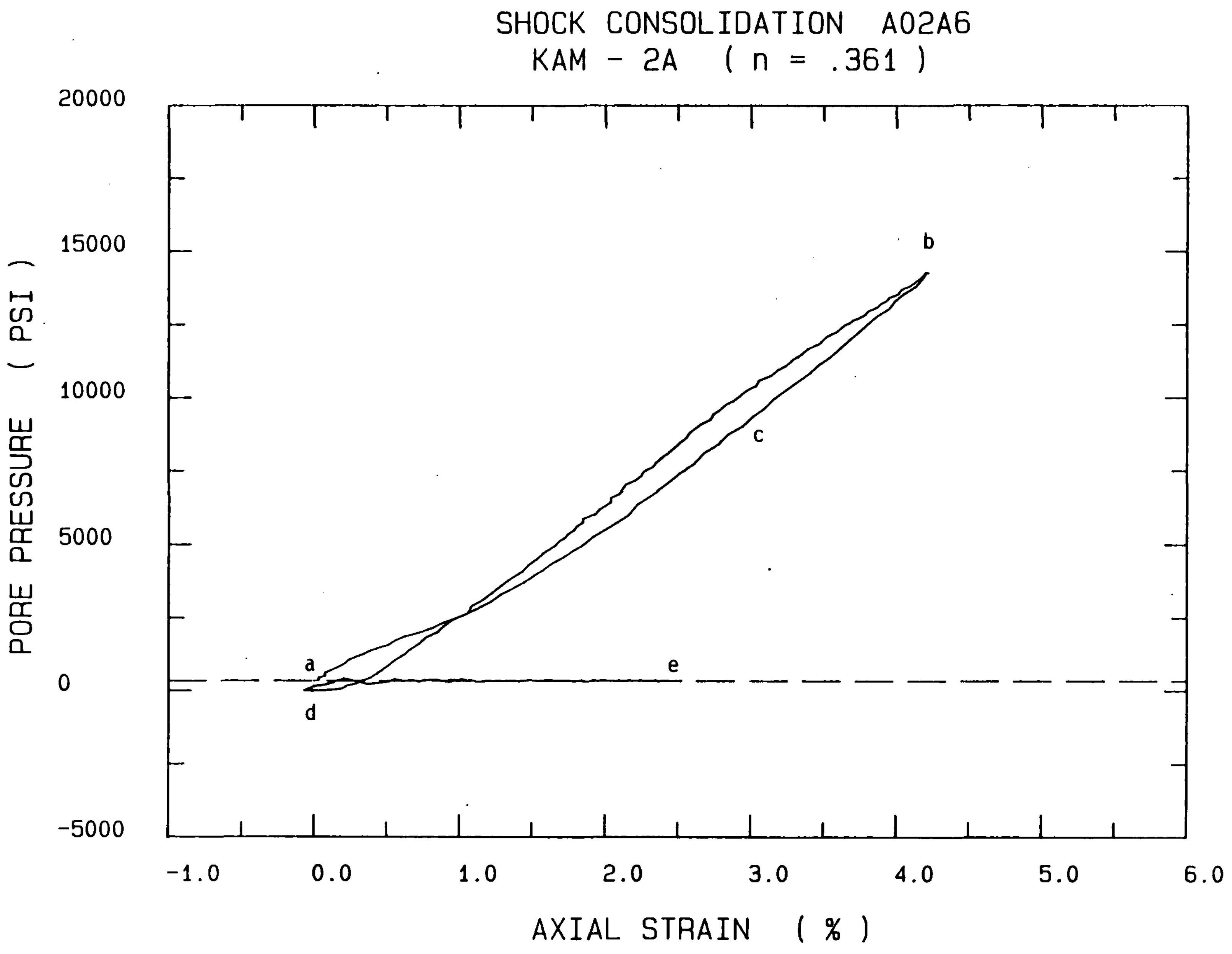


Figure 3.5. Pore pressure vs. axial strain for a cemented limestone sample.

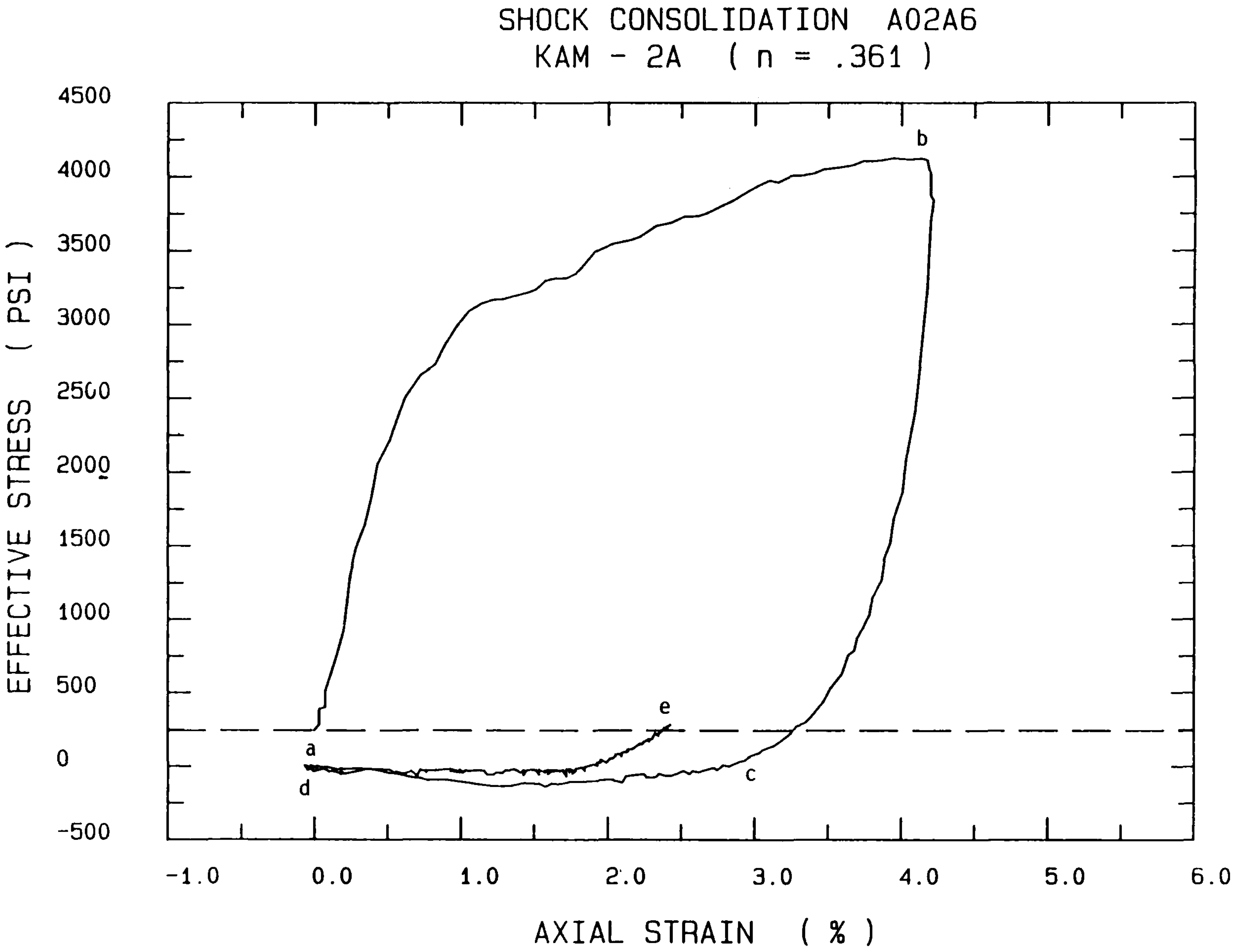


Figure 3.6. Effective stress vs. axial strain for a cemented limestone sample.

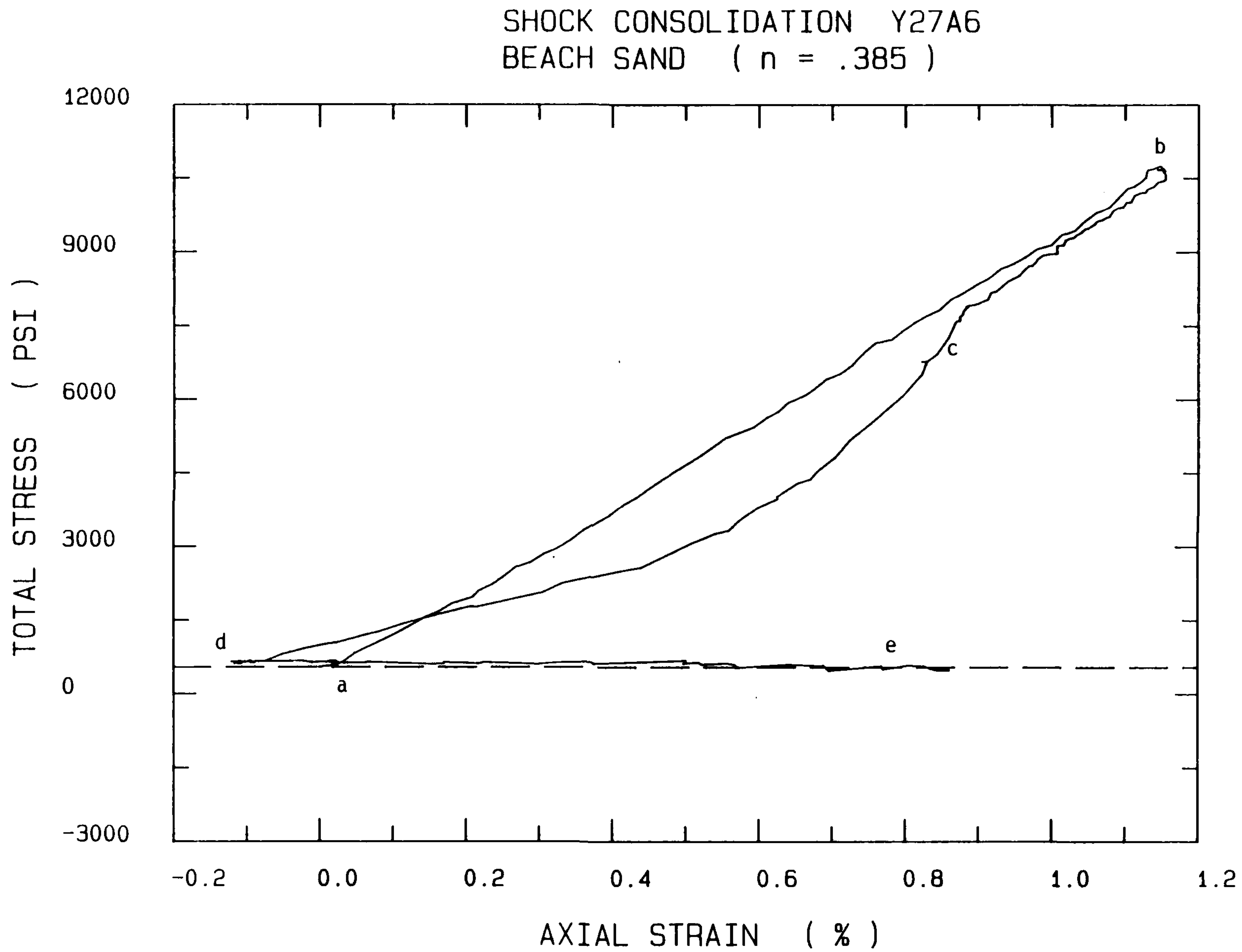


Figure 3.7. Axial stress vs. axial strain, K_o triaxial shock consolidation on a beach sand.

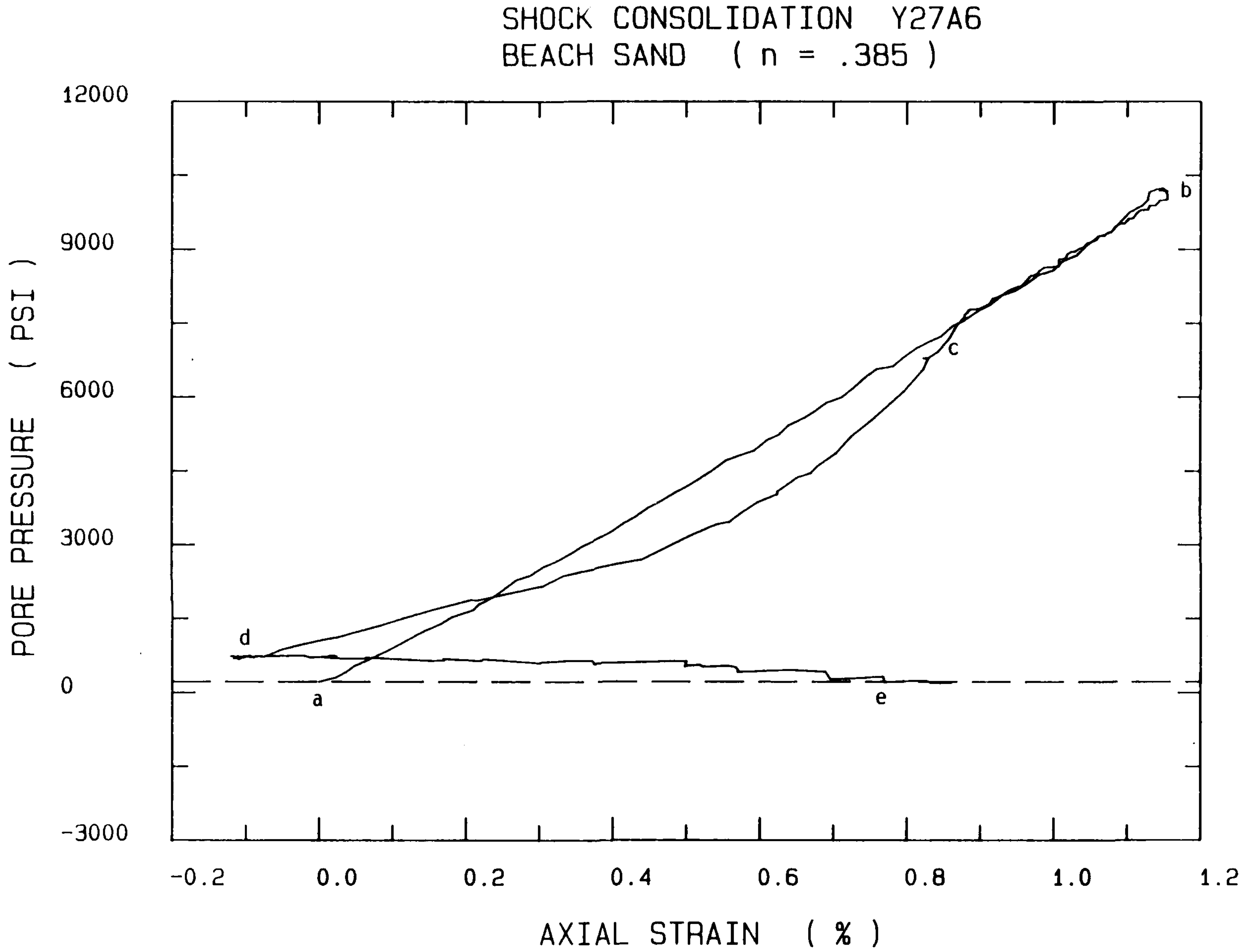


Figure 3.8. Pore pressure vs. axial strain, K_0 triaxial shock consolidation on a beach sand.

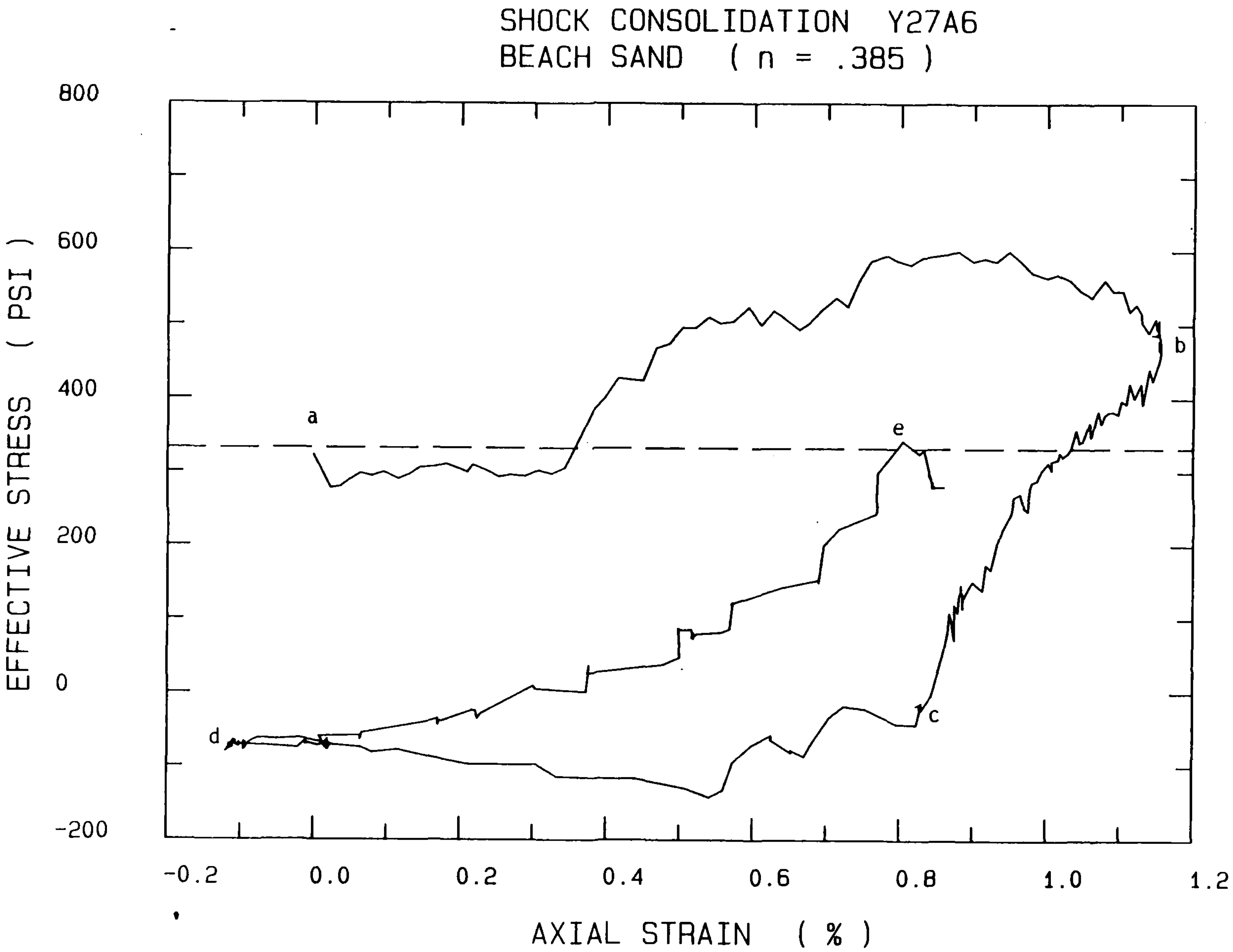


Figure 3.9. Effective stress vs. axial strain, K_0 triaxial shock consolidation on a beach sand.

SECTION 4

DESIGN AND EVALUATION OF A DYNAMIC FLOW TEST DEVICE

4.1 INTRODUCTION

As described in Section 7, a key component of multiphase modeling is the equation of motion of the pore fluid within the porous skeleton. Nearly all previous work in this area is founded on work by Biot (1956, 1962a, 1962b). For blast loadings, however, Biot's work is probably inadequate because it assumes laminar flow and idealized simple pore geometries (i.e. circular and flat ducts). For real porous media and flow rates which are in the transition and turbulent flow regimes, a more comprehensive modeling formulation will be required. An interim formulation which incorporates Biot's theoretical work and Ward's (1964) empirical results is presented in Appendix E.

This section describes the design and fabrication of a servo-controlled dynamic flow device which has been built and is currently being evaluated. With this device we hope to generate an empirically based fluid friction model which will be generally applicable in all three flow regimes and which will supersede the formulation presented in Appendix E.

4.2 DESIGN PROCEDURES

Design criteria for the dynamic flow test device included sample size requirements, pressure and pressure gradient specifications, control requirements and data acquisition specifications. It was determined that the device must be capable of testing samples with diameters up to 2.0 in and lengths up to 9 in. Other capabilities which were specified are: the ability to pressurize the fluid to 5000 psi (yielding flow gradients of up to 6700 psi/ft on a 9 in sample), the ability to control and measure the rate of flow through the sample, the ability to measure the pressure and temperature on the high and low pressure sides of the sample, and the ability to change either the sample or the fluid easily.

Based on the listed criteria a device was designed and constructed which met all the requirements. A schematic of the device is shown in Figure 4.1. The device was constructed from components of a 6 in hydraulic cylinder w)ich were modified to contain a sample holder and the pressure and temperature sensors.

4.3 DESIGN ANALYSIS AND RESEARCH PLANS

The test device has recently been completed and a complete series of evaluation tests is not underway. Initially, a series of tests is being performed on a cylindrical duct to compare the output of our device with published values of frictional resistance under steady state conditions in the laminar, transition and turbulent flow regimes. Raw output data from a typical test are shown in Figures 4.2 through 4.4. Pressure gradients can be calculated using the measured pressures and the sample length. The results will be plotted as a function of flow rate through the duct.

After completing the calibration series of tests, various other samples will be tested including noncylindrical ducts, ball bearings and various soil and rock samples. These samples will be tested in both the static and dynamic modes and the results analyzed and an updated model formulation for fluid friction prepared.

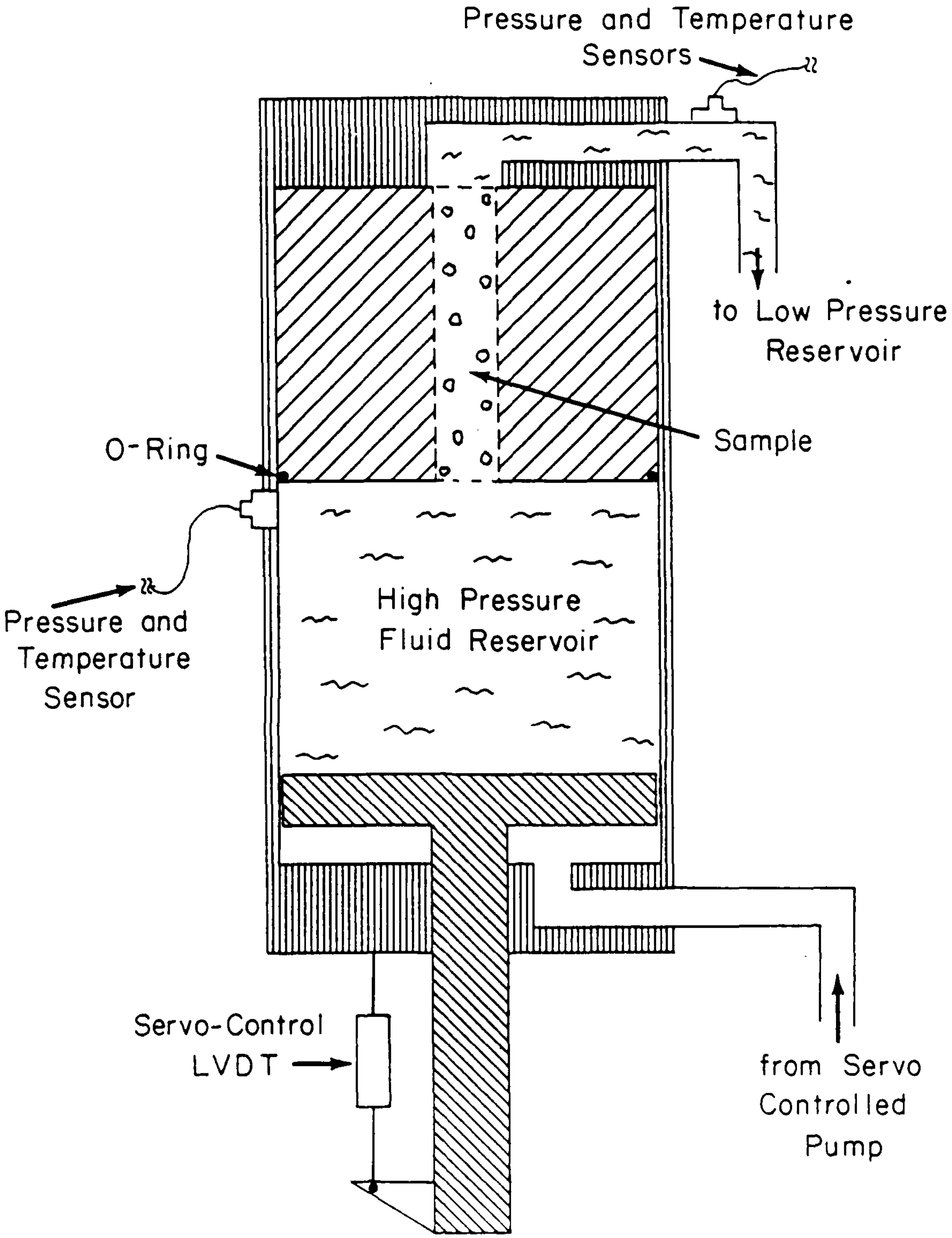


Figure 4.1. Schematic section view of servo controlled fluid flow device.

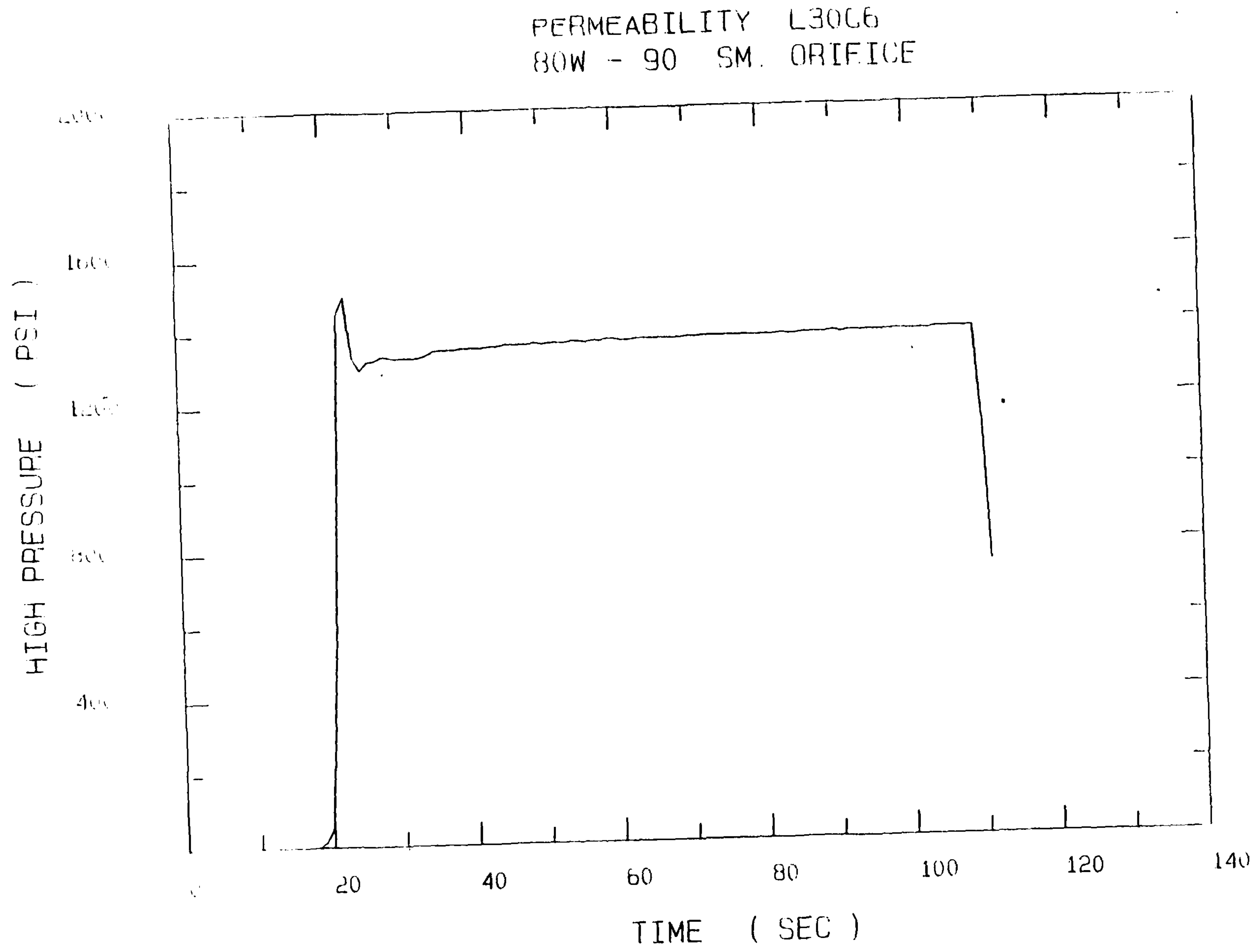


Figure 4.2. Raw data - pressure time history in high pressure fluid reservoir.

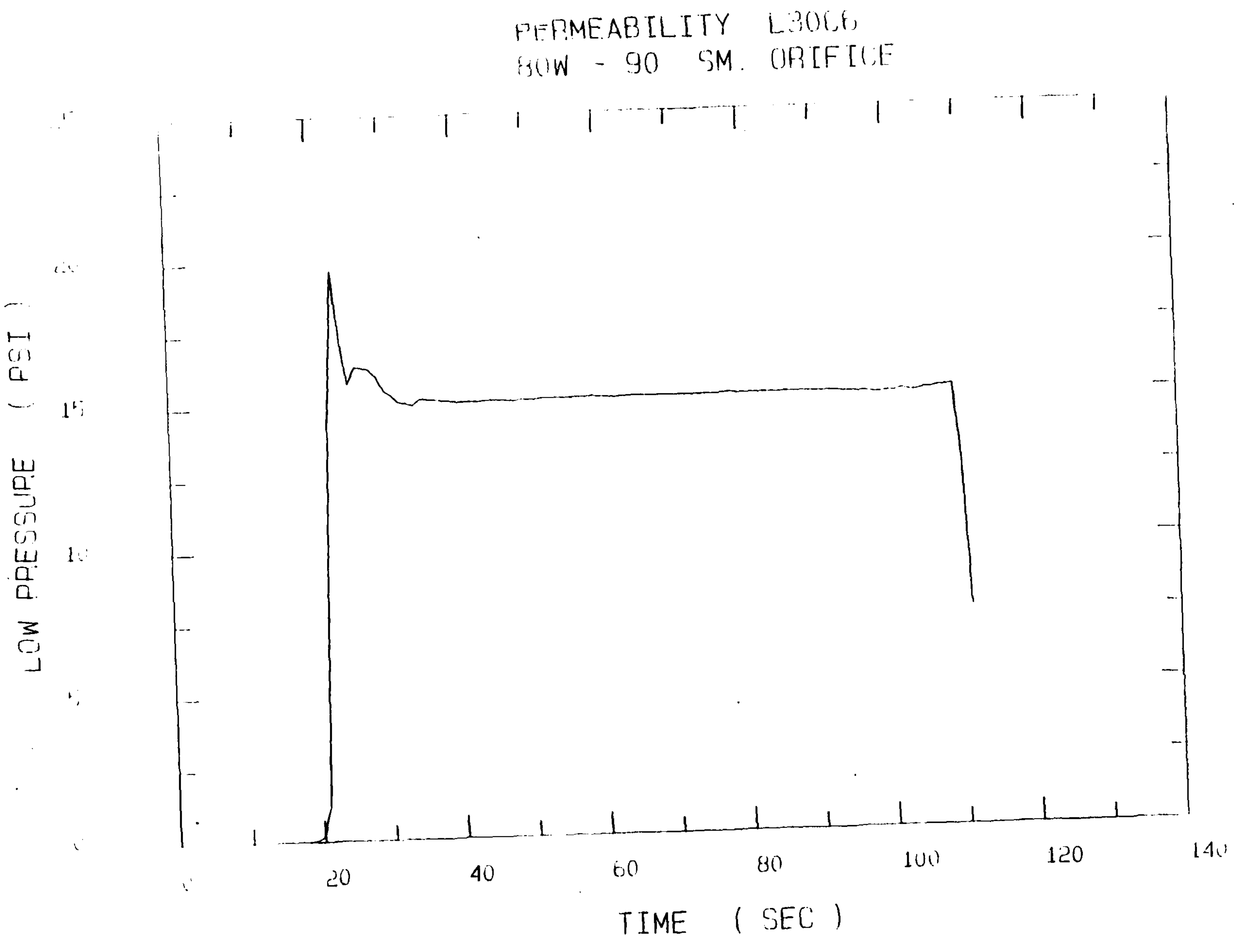


Figure 4.3. Raw data - pressure time history in low pressure fluid reservoir.

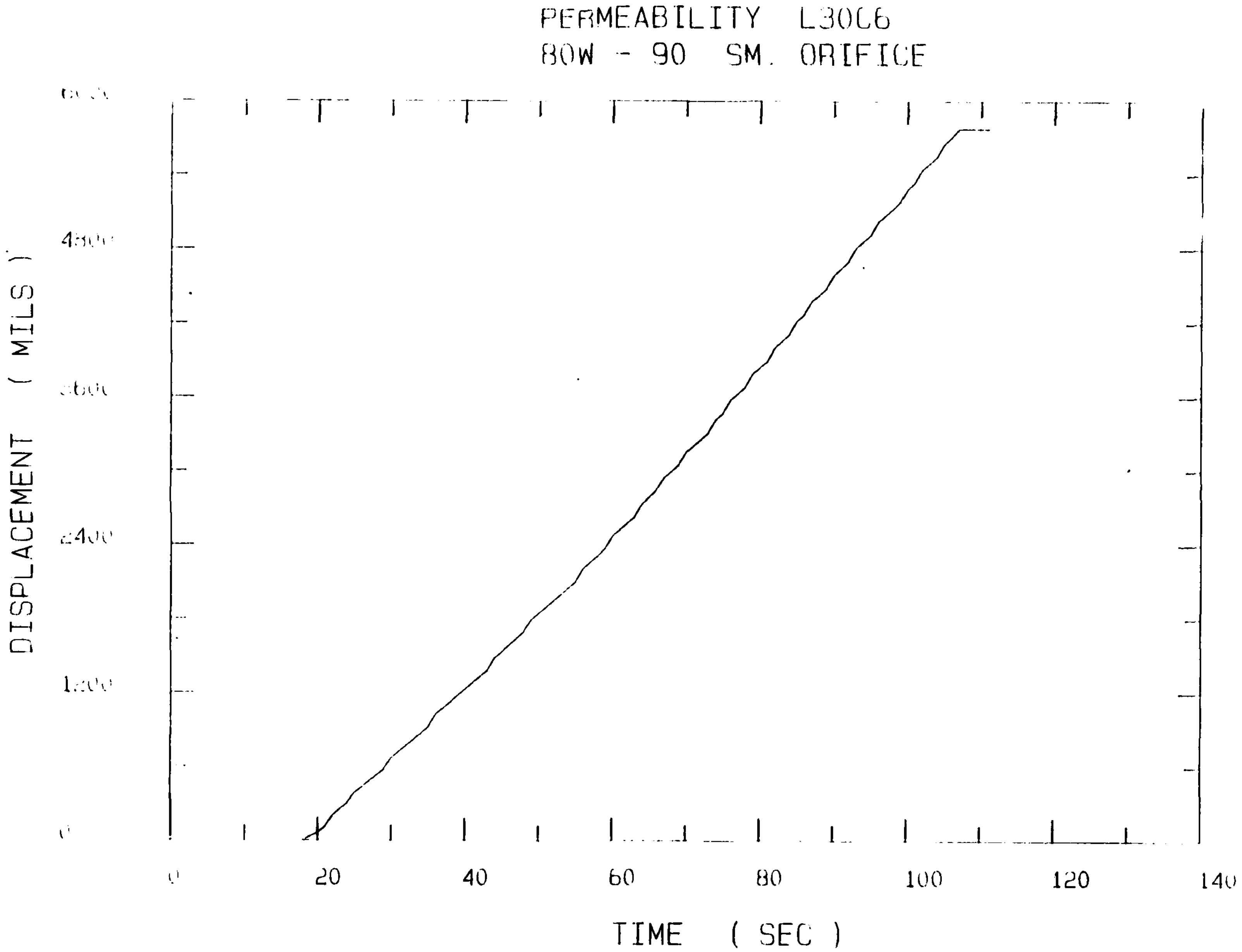


Figure 4.4. Raw data - displacement of 6" piston into high pressure reservoir.

SECTION 5

COMPUTATION OF UNDRAINED MATERIAL RESPONSE IN HYDROSTATIC AND UNIAXIAL COMPRESSION

5.1 INTRODUCTION

Multiphase models require a complete description of the fluid and solid phases of the components which make up the bulk material. In the case of the pore fluid components this includes a description of the pressure volume characteristics. In the case of the solid skeleton component this includes a description of the grain compressibility and a description of the skeleton response to the appropriate loading conditions. The grain compressibility is used in establishment of coupling parameters between the pore fluid and the solid skeleton. Because of this coupling, the skeleton properties cannot be extracted directly from undrained tests, nor can the drained skeleton properties be simply incorporated into a single phase model of the bulk material behavior.

Because multiphase models are not yet routinely used in numerical calculations, equivalent single phase total stress response must frequently be provided based on the properties obtained to support two-phase calculations. To meet this need we have developed analytical procedures to compute the undrained response of fully saturated materials in hydrostatic and uniaxial compression. These procedures are implemented in three simple computer programs described in Sections 5.3 and 5.4. Constitutive relationships for the individual material components described in Section 5.2 are used as inputs to these programs.

5.2 COMPONENT CONSTITUTIVE RELATIONSHIPS

5.2.1 Introduction

Three constitutive relationships are required to describe the undrained bulk behavior of saturated porous materials using a fully coupled material model. These are:

- the compressibility characteristics of the solid grains which make up the material skeleton;

- the compressibility characteristics of the pore fluid; and

- the deformation characteristics of the material skeleton.

These relationships are discussed individually in the following three subsections. At this time the unloading relationships are not complete; these will be included in next year's effort.

5.2.2 Grain Compressibility

The solid grain compressibility model is depicted schematically in Figure 5.1. Though, as shown in Section 2, most materials have linear pressure volume response at kilobar pressure levels, the model handles nonlinear response if needed. The hydrostatic pressure, p_G, applied to the solid grains results in a volumetric strain, ϵ_G. The volume strain is expressed in terms of engineering strain according to

$$\epsilon_G = \frac{V_{Go} - V_G}{V_{Go}} \qquad (5\text{-}1)$$

where

V_{Go} = initial volume of the solid grains

V_G = volume of the solid grains at pressure p_G

The tangential bulk modulus, K_G, is defined as the slope of the pressure-volume strain curve at a specific pressure level and is given by

$$K_G = \frac{\Delta p_G}{\Delta \epsilon_G} \qquad (5\text{-}2)$$

Where available, bulk modulus of the solid grains as a function of pressure can be obtained directly from laboratory data such as those presented in Section 2. In cases where data are not available, moduli can be obtained using empirical relationships from the literature such as those given by Bridgman (1931) as

$$\epsilon_G = a \times 10^{-7}\, p_G - b \times 10^{-12}(p_G)^2$$

where a and b are empirically derived constants for various elements and compounds. For example, calcium at 30° C has values of a = 56.97 and b = 47.2 where p_G is expressed in units of kg/cm^2.

For our example calculations we fit data for quartz with the following relationships:

$$K_G = 344.\ (\text{Kbars})\ \text{for}\ p_G < 1.36\ \text{Kbars}$$

$$K_G = 253.\ p_G\ (\text{Kbars})\ \text{for}\ p_G \geqslant 1.36\ \text{Kbars}$$

For most problems with pressures less than 10 Kbars a constant grain modulus is usually sufficient.

5.2.3 Compressibility of Pore Water

The pore water compressibility model is also depicted schematically in Figure 5.1. Application of pore pressure, π, results in a volume strain of ϵ_w, in the pore water where

$$\epsilon_w = \frac{V_{wo} - V_w}{V_{wo}} \qquad (5\text{-}3)$$

where

V_{wo} = initial volume of pore water, and

V_w = volume of pore water at a specific pore pressure, π.

The tangential bulk modulus, K_w, is defined as the slope of the pore pressure-volume strain curve at a specific pore pressure level, i.e.

$$K_w = \frac{\Delta\pi}{\Delta\epsilon_w} \qquad (5\text{-}4)$$

Compressibility models for plain water and sea water have been developed for use in the computerized analytic procedures based on a variety of data from the literature. Britt's fit (1985) to the data of Wilson and Bradley (1966 and 1968) gives the specific volume of plain water and sea water as a function of temperature, pressure and salinity as

$$V_S = 0.70200 + \frac{100(17.5273+0.1101T-0.000639T^2-0.039986S-0.000107TS)}{(\pi a + 5880.9 + 37.592T - 0.34395T^2 + 2.2524S)} \qquad (5\text{-}5)$$

where the absolute pressure, πa, is expressed in bars, the temperature T is in degrees C, and the salinity, S, is in parts per thousand. The equation is accurate to 0.00013 cm^3/g for pressures from 0 to 1500 bars and temperatures from 0 to 40°C.

Table 5.1 shows values of the specific volume of fresh water and sea water for a range of pressures from 0 to 800 Kbars. The values for pressures from 0 to 1.5 Kbars were computed from Equation 5-5 for a temperature of 25°C and salinities of 0 and 32 parts per thousand. The latter value is a typical salinity for sea water. Above 1.5 Kbars Britt's recommended Hugoniot for fresh water was used. This is based on data by Grindley and Lind (1971) in the 1 to 8 Kbar regime, Bridgman (1912) in the 8 to 12.5 Kbar regime, Rice and Walsh (1957) in the 12.5 to 450 Kbar regime and Skidmore and Morris above 450 Kbars.

Values for the specific volume of sea water above 1.5 Kbars were estimated from Britt's fits to Adam's data (1931) which compare the volume strain in fresh water to that in salt water with a salinity of 32 parts per thousand at 25°C. Figure 5.2 shows our recommended relationship between the volume strains in fresh water and sea water for pressures above 1.5 Kbars, given by

$$\frac{\epsilon_{wf}}{\epsilon_{ws}} = 1 + 0.074\ \pi^{-0.20} \qquad (5\text{-}6)$$

where ϵ_{wf} is the volume strain in fresh water and ϵ_{ws} is the volume strain in sea water. The pressure, π, is expressed in Kbars. Equation 5-6 was used to compute the specific volumes and volume strains in the sea water shown in Table 5.1 for pressures above 1.5 Kbars.

The specific volumes of Table 1 were used to compute the volume strains in engineering strain according to

$$\epsilon_W = \frac{V_{so} - V_s}{V_{so}} \qquad (5\text{-}7)$$

where V_{so} is the specific volume at a pressure of zero (1 atmosphere absolute) and V_s is the specific volume at the pressure of interest. The bulk moduli of Table 5.1 are tangent moduli computed over each pressure increment according to Equation 5-4.

5.2.4 Skeleton Compressibility

The skeleton compressibility for any material of interest must be defined from hydrostatic and uniaxial strain drained loadings. As shown in the schematic view of Figure 5.1, application of either axial effective stress, σ' (for the uniaxial loading), or mean effective pressure, p' (for the hydrostatic loading), results in an effective volume strain in the skeleton, ϵ'. The effective volumetric strain is defined as

$$\epsilon' = \frac{V_o - V}{V_o} \qquad (5\text{-}8)$$

where

V_o is the initial apparent volume of the skeleton, and

V is the apparent volume of the skeleton at either a specific axial effective stress, σ', or at a specific effective mean pressure, p'.

For the hydrostatic loading the bulk modulus, K_s, is defined as

$$K_s = \frac{\Delta p'}{\Delta \epsilon'} \qquad (5\text{-}9)$$

For the uniaxial strain loading the constrained modulus, M_s, is defined as

$$M_S = \frac{\Delta\sigma'}{\Delta\epsilon'} \qquad (5\text{-}10)$$

Examples of effective stress-strain curves in the uniaxial loading for uncemented, weakly cemented and well cemented Enewetak carbonates (Blouin and Timian, 1986) are shown in Figure 5.3. In the cemented materials there is a sharp initial increase in stress until the cementation breaks down (at about 1000 psi and 5000 psi in the weakly and strongly cemented materials, respectively). The breakdown of cementation is followed by a much softer response until the void volume has been significantly reduced. The hydrostatic response is very similar to the uniaxial response in the corresponding materials. These skeleton response curves are used in the example calculations in Section 5.4.3.

5.3 MIXTURE MODEL

5.3.1 Introduction

As described in Blouin and Kim (1984), the simplest model for representing bulk behavior of saturated porous material is the so called mixture model. This model, originally developed by Wood (1930), treats the solid grain/pore water as a mixture having no effective stress. The composite material is thus treated as a heavy fluid having a density which is governed by the porosity. The bulk material response represents the lowest possible bound for bulk modulus. In general it gives a reasonable approximation of the response of saturated soils, but, as shown in Blouin and Kim (1984), it becomes deficient in cemented and dense porous materials.

In the following subsections we develop formulations and computer codes which model the bulk response of saturated mixtures based on incremental approximations of the actual pressure volume response of the sold grains and pore water. These formulations are derived in terms of both engineering strain and natural, or true, strain.

5.3.2 Mixture Model Based on Engineering Strain

In developing the mixture model, the pore water and solid grain responses must be combined using mixture theory.

5.3.2.1 Bulk Modulus of Pore Water

The change in volume strain due to a pressure increment, $\Delta\pi_i$, is given by

$$\Delta\epsilon_{wi} = \frac{\Delta V_{wi}}{V_{wo}} \tag{5-11}$$

where ΔV_{wi} is the change in pore water volume.

The tangent bulk modulus over the pressure increment, $\Delta\pi_i$, is given by

$$K_{wi} = \frac{\Delta\pi_i}{\Delta\epsilon_{wi}} \tag{5-12}$$

The initial pore water volume is given by

$$V_{wo} = n_o V_o \tag{5-13}$$

where n_o is the initial porosity.
Combining equations 5-11 through 5-13 yields

$$\Delta V_{wi} = \frac{n_o}{K_{wi}} \Delta\pi_i V_o \tag{5-14}$$

5.3.2.2 Bulk Modulus of Solid Grains

The change in volume strain of the solid grains is

$$\Delta\epsilon_{Gi} = \frac{\Delta V_{Gi}}{V_{Go}} \tag{5-15}$$

where ΔV_{Gi} is the volume change of the solid grains. The tangent bulk modulus of the solid grains is given by

$$K_{Gi} = \frac{\Delta p_{Gi}}{\Delta \epsilon_{Gi}} \qquad (5\text{-}16)$$

where Δp_{Gi} is the change in hydrostatic pressure acting on the solid grains. Finally the initial volume of the solid grains is given by

$$V_{Go} = (1 - n_o)V_o \qquad (5\text{-}17)$$

Combining 5-15 through 5-17 gives

$$\Delta V_{Gi} = \frac{(1 - n_o)}{K_{Gi}} \Delta P_{Gi} V_o \qquad (5\text{-}18)$$

5.3.2.3 Undrained Bulk Modulus of Mixture

Since there is no effective stress, the total applied stress Δp_i is equal to the pore pressure which is also equal to the hydrostatic stress applied to the grains. Thus

$$\Delta \pi_i = \Delta p_i \qquad (5\text{-}19a)$$

$$\Delta p_{Gi} = \Delta p_i \qquad (5\text{-}19b)$$

Also the change in current total volume of the mixture equals the sum of the volume changes of the components;

$$\Delta V_i = \Delta_{Gi} + \Delta V_{wi} \qquad (5\text{-}20)$$

Substituting Equations 5-14 and 5-18 into 5-20 yields

$$\Delta V_i = \frac{(1 - n_o)}{K_{Gi}} \Delta p_{Gi} V_o + \frac{n_o}{K_{wi}} \Delta \pi_i V_o \qquad (5\text{-}21)$$

Incorporating the relationships of Equations 5-19 into 5-21 yields

$$\frac{\Delta V_i}{V_o} = \left[\frac{1 - n_o}{K_{Gi}} + \frac{n_o}{K_{wi}}\right] \Delta p_i \qquad (5\text{-}22)$$

The change in total strain, $\Delta \epsilon_i$, is given by

$$\Delta\epsilon_i = \frac{\Delta V_i}{V_o} \tag{5-23}$$

and the undrained bulk modulus of the mixture, K_{mi}, is defined as

$$K_{mi} = \frac{\Delta p_i}{\Delta\epsilon_i} \tag{5-24}$$

Combining Equations 5-22 through 5-24 yields

$$\boxed{K_{mi} = \frac{K_{Gi}\ K_{wi}}{K_{wi} + n_o(K_{Gi} - K_{wi})}} \tag{5-25}$$

Thus, the bulk modulus of the mixture at step i is a function of the bulk moduli of the solid grains and pore water over the pressure increment $\Delta\pi_i$ and the initial porosity.

5.3.2.4 Algorithm for Computation of Undrained Behavior of Mixture

Using Equations 5-24 and 5-25 the computational algorithm presented in Appendix A.1 was prepared. The program logic, shown schematically in the flow chart of Figure 5.4, follows the usual step by step incremental calculational procedure for non-linear analysis. An example calculation using the above algorithm is presented in Appendix A.2.

5.3.3 Mixture Model Based on True Strain

Frequently, computational algorithms utilize natural or true strain rather than engineering strain to describe constitutive relationships. In order to support such algorithms the undrained mixture response must be formulated in terms of true strain.

True or natural strain is defined by

$$\bar{\epsilon}_i = \ln \frac{l_o}{l_i} \tag{5-26}$$

where $\bar{\epsilon}_i$ is the true strain, l_i the current length and l_o the initial sample length. Equation 5-26 is applicable when compressive strains are defined as positive. Equation 5-26 may also be written as

$$\bar{\epsilon}_i = -\ln(1 - \epsilon_i) \qquad (5\text{-}27)$$

where ϵ_i is the engineering strain given by

$$\epsilon_i = \frac{l_i}{l_o} \qquad (5\text{-}28)$$

True strain is continually updated such that application of a given increment of true strain represents a given percentage change in dimension independent of the strain magnitude when the increment is applied. For example, a 5% increment of true strain represents a 5% change in dimension irrespective of the initial strain at the start of the increment. In terms of engineering strain a 5% strain increment is a fixed change in dimension. It represents a 5% change in dimension only when measured from the initial sample dimension l_o. For a sample undergoing compression, a 5% change in engineering strain represents more than a 5% change in length whenever this strain increment is applied after the start of deformation. Engineering strain and true strain are nearly equal for small strains, but at larger strains the difference becomes significant. Figure 5.5 shows a plot of true strain as a function of engineering strain which illustrates the increasing divergence at larger strain values.

5.3.3.1 Undrained Bulk Mixture Modulus Based on True Strain

The derivation of the undrained response of the bulk mixture based on true strain is similar to that presented under subsection 5.3.2 for engineering strain. The bulk modulus of the pore water in terms of true strain, K_{wi}, over the pressure increment, $\Delta\pi_i$, is given by

$$\bar{K}_{wi} = \frac{\Delta\pi_i}{\Delta\bar{\epsilon}_{wi}} \qquad (5\text{-}29)$$

where $\Delta\bar{\epsilon}_{wi}$ is the change in the true volume strain which is expressed as

$$\Delta\bar{\varepsilon}_{wi} = \frac{\Delta V_{wi}}{\bar{V}_{wi}} \qquad (5\text{-}30)$$

The bulk modulus of the solid grains in terms of true strain, $\bar{K}_{Gi}$, is given by

$$\bar{K}_{Gi} = \frac{\Delta p_{Gi}}{\Delta\bar{\varepsilon}_{Gi}} \qquad (5\text{-}31)$$

where $\Delta\bar{\varepsilon}_{Gi}$ is the change in true volume strain which is expressed as

$$\Delta\bar{\varepsilon}_{Gi} = \frac{\Delta V_{Gi}}{\bar{V}_{Gi}} \qquad (5\text{-}32)$$

The current porosity, n_i, is defined as

$$n_i = \frac{V_{wi}}{\bar{V}_i} \qquad (5\text{-}33)$$

The undrained bulk modulus of the mixture in terms of true strain, $\bar{K}_{mi}$, is given by

$$\bar{K}_{mi} = \frac{\Delta p_i}{\Delta\bar{\varepsilon}_i} \qquad (5\text{-}34)$$

Combining Equations 5-29 through 5-34 and Equations 5-19 and 5-20 in the same manner as for engineering strain yields

$$\boxed{\bar{K}_m = \frac{\bar{K}_{Gi}\,\bar{K}_{wi}}{\bar{K}_{wi} + n_i(\bar{K}_{Gi} - \bar{K}_{wi})}} \qquad (5\text{-}35)$$

Thus the bulk modulus of the mixture in terms of true strain at step i is a function of the bulk moduli of the solid grains and pore water expressed in terms of true strain and the current porosity. Note that the current porosity is used in the expression for the bulk modulus in terms of true strain while the initial porosity is used to express bulk modulus in terms of engineering strain.

5.3.3.2 Algorithm for Computation of Undrained Behavior of Mixture in Terms of True Strain

A second computational algorithm to compute the undrained response of the mixture based on true strain was developed from Equations 5-34 and 5-35. This is presented in Appendix B.1. The program logic is the same as that presented for the engineering strain program in Figure 5.4. The same example problem solved in Appendix A.2 is solved in terms of true strain in Appendix B.2. The very minor differences between the results of the two calculations are due to the incremental approximations to the pressure deformation response of the solid grains and water.

5.4 FULLY COUPLED MODEL

5.4.1 Introduction

Blouin and Kim (1984) describe three models which treat the effective stress response of the saturated soil or rock skeleton. The resistance of the soil skeleton to compressive loadings is in addition to the resistance of the solid-water mixture described in Section 5.3. The three models, termed the decoupled, partially coupled, and fully coupled models, differ in their level of sophistication.

The simplest model is the decoupled model in which the soil or rock skeleton acts in parallel to the solid-water mixture. The resultant compressive moduli are simply the sum of the mixture modulus plus the appropriate skeleton modulus. The partially coupled model includes the strain in the skeleton which results from compression of the individual grains by the pore pressure. The most sophisticated model is the fully coupled, which, in addition to the assumptions for the partially coupled model, includes volume change in the soil-water mixture due to effective stress acting on the solid grains.

In this section we derive formulations for undrained response due to hydrostatic and uniaxial strain loadings using the fully coupled material model. These formulations are based on engineering strain. They represent the most sophisticated means of predicting undrained response from drained skeleton properties and are valid even for strongly cemented porous materials and at high strain and effective stress levels. These formulations will also provide a means of verifying the multiphase dynamic analysis program, MPDAP.

5.4.2 Hydrostatic Compression

In the fully coupled model the pore water pressure acting on the solid grains results in a volume decrease in the entire material skeleton. In addition, the effective stress in the soil grains results in further compression of the individual grains and an additional volume decrease in the soil-water mixture not accounted for in the less sophisticated models.

At the specified step, i, the change in the bulk volume strain, $\Delta\epsilon_i$, is defined as

$$\Delta\epsilon_i = \frac{\Delta V_i}{V_o} \tag{5-36}$$

Two components contribute to the net volume change;

$$\Delta V_i = \Delta V_{mi} + \Delta V'_{Gi} \tag{5-37}$$

where ΔV_{mi} is the volume change in the solid-water mixture due to the pore water pressure and $\Delta V'_{Gi}$ is the volume change in the solid grains due to the effective stress change, $\Delta p'$. From Equations 5-23 and 5-24,

$$\Delta V_{mi} = \frac{\Delta\pi_i}{K_{mi}} V_o \tag{5-38}$$

where K_{mi} is the mixture mcdulus given by equation 5-25. The average change in grain effective stress, $\Delta\sigma'_{Gi}$, is given by

$$\Delta\sigma'_{Gi} = \frac{\Delta p'_i}{1-n_i} \tag{5-39}$$

The change in volume strain in the solid grains due to the effective stress change is given by

$$\Delta\epsilon'_{Gi} = \frac{\Delta\sigma'_{Gi}}{K_{Gi}} \qquad (5\text{-}40)$$

where

$$\Delta\epsilon'_{Gi} = \frac{\Delta V'_{Gi}}{V_{Go}} \qquad (5\text{-}41)$$

Substituting Equation 5-39 into 5-40 yields

$$\Delta\epsilon'_{Gi} = \frac{\Delta p'_i}{(1-n_i)K_{Gi}} \qquad (5\text{-}42)$$

By definition,

$$1 - n_i = \frac{V_{Gi}}{V_i} \qquad (5\text{-}43)$$

Substituting Equation 5-43 into 5-42 gives

$$\Delta\epsilon'_{Gi} = \frac{\Delta p'_i\ V_i}{V_{Gi}\ K_{Gi}} \qquad (5\text{-}44)$$

From Equations 5-41 and 5-44

$$\Delta V'_{Gi} = \frac{\Delta p'_i V_i V_{Go}}{V_{Gi}\ K_{Gi}} \qquad (5\text{-}45)$$

Substitution of Equations 5-38 and 5-45 into Equation 5-37 gives

$$\Delta V_i = \frac{\Delta\pi_i}{K_{mi}} V_o + \frac{\Delta p'_i\ V_i\ V_{Go}}{V_{Gi}\ K_{Gi}} \qquad (5\text{-}46)$$

By definition

$$V_{Go} = (1 - n_o)V_o \qquad (5\text{-}47)$$

which substituted into Equation 5-46 yields the net volume change as,

$$\Delta V_i = V_o \left[\frac{\Delta\pi_i}{K_{mi}} + \frac{(1-n_o)\Delta p'_i V_i}{V_{Gi}\ K_{Gi}}\right] \qquad (5\text{-}48)$$

Combining Equations 5-36 and 5-43 with 5-48 gives

$$\Delta\epsilon_i = \frac{\Delta\pi_i}{K_{mi}} + a\ \frac{\Delta p'_i}{K_{Gi}} \qquad (5\text{-}49)$$

where

$$a = \frac{1-n_o}{1-n_i} \tag{5-50}$$

The other strain compatability equations are

$$\Delta\epsilon_i = \frac{\Delta p_i}{K_{fi}} \tag{5-51}$$

where K_{fi} is the bulk modulus for the fully coupled model, and

$$\Delta\epsilon_i = \frac{\Delta p'_i}{K_{si}} + \frac{\Delta\pi_i}{K_{Gi}} \tag{5-52}$$

Equation 5-52 represents the total strain in the skeleton due to the effective stress plus the compression of the grains by the pore pressure.

Equating Equations 5-49 and 5-52 yields

$$\Delta p'_i = \frac{K_{si}\ (K_{Gi}-K_{mi})}{K_{mi}(K_{Gi}-aK_{si})}\ \Delta\pi_i \tag{5-53}$$

The effective stress law is given by

$$\Delta p_i = \Delta\pi_i + \Delta p'_i \tag{5-54}$$

Combining Equations 5-25, 5-49, 5-51, 5-53 and 5-54 with a bit of manipulation yields

$$\boxed{K_{fi} = K_{si} + \frac{K_{Gi} - aK_{si}}{1 + n_o\,\dfrac{K_{Gi}\ (K_{Gi}-K_{wi})}{K_{wi}(K_{Gi}-aK_{si})}}} \tag{5-55}$$

Equation 5-55 is similar to the linear fully coupled equation developed by Blouin and Kim (1984) except for the inclusion of the parameter, a. The parameter, a, decreases from unity at the start of compression, corresponding to the linear assumption in Blouin and Kim (1984). For the case where the skeleton modulus is zero, Equation 5-55 degenerates to the mixture modulus given by Equation 5-25.

The pore pressure response as a function of the total strain increment is obtained by combining Equations 5-51, 5-52 and 5-54,

$$\boxed{\Delta\pi_i = \frac{K_{fi}-K_{si}}{1 - \dfrac{K_{si}}{K_{Gi}}} \Delta\epsilon_i} \tag{5-56}$$

5.4.3 Algorithm for Fully Coupled Hydrostatic Response of Saturated Materials

A computational algorithm to compute the fully coupled response of saturated porous media to undrained hydrostatic loadings is presented in Appendix C.1. The flow chart for this program is shown in Figure 5.6. Inputs to the program include the drained skeleton response to hydrostatic loading as well as the solid grain and water pressure-volume characteristics.

Two example calculations are presented in Appendix C.2, one for a well cemented saturated coral and the other for saturated beach sand. Skeleton behavior of both materials is similar to that shown for the uniaxial loadings of Figure 5.3.

5.4.4 Uniaxial Compression

The development of the constitutive equations for uniaxial strain using the fully coupled model is similar to that for the hydrostatic loading. The overall volume strain increment is defined as

$$\Delta\epsilon_i = \frac{\Delta\sigma_{ai}}{M_{fi}} \tag{5-57}$$

where

$\Delta\sigma_{ai}$ is the total axial stress increment, and

M_{fi} is the undrained constrained modulus

The development follows the procedures described by Blouin and Kim (1984) for the uniaxial fully coupled loading with the same modification of the Equation for constrained modulus as derived in Section 5.4.2 for the bulk modulus. Thus, the undrained constrained modulus is given by

$$\boxed{M_{fi} = M_{si} + \frac{K_{Gi}-aK_{si}}{1 + n_o \dfrac{K_{Gi}(K_{Gi}-K_{wi})}{K_{wi}(K_{Gi}-aK_{si})}}} \tag{5-58}$$

The incremental pore pressure response is given by

$$\boxed{\Delta\pi_i = \frac{M_{fi}-M_{si}}{1 - \dfrac{K_{si}}{K_{Gi}}} \Delta\epsilon_i} \tag{5-59}$$

where

M_{si} is the drained constrained modulus of the skeleton.

5.4.5 Algorithm for Fully Coupled Uniaxial Response of Saturated Materials

A computational algorithm to compute the fully coupled undrained response to uniaxial strain loadings is presented in Appendix D.1. The program flow chart is shown in Figure 5.7. Program inputs include the skeleton response to uniaxial drained loading and the pressure volume response of the pore water and the solid grains.

Example calculations of the uniaxial undrained response of well cemented coral and beach sand are presented in Appendix D.2. The uniaxial skeleton response for these two materials is shown in Figure 5.3.

Table 5.1. Compression properties of fresh water and sea water.

	Fresh Water			Sea Water		
π Pressure (bars)	V_s Specific Volume (cm^3/g)	ϵ_v Volume Strain	K_{wf} Bulk Modulus Salinity = 0 parts/ thousand (bars)	V_s Specific Volume (cm^3/g)	ϵ_v Volume Strain	K_{ws} Bulk Modulus Salinity = 32 parts/ thousand (bars)
0.	1.00291		22188.	.97922		23767.
50.	1.00065	.00225	22188.	.97716	.00210	23767.
100.	.99842	.00448	22452.	.97513	.00418	24076.
150.	.99623	.00666	22929.	.97313	.00622	24519.
200.	.99407	.00881	23209.	.97116	.00823	24863.
250.	.99194	.01094	23494.	.96922	.01021	25224.
300.	.98984	.01303	23900.	.96731	.01216	25605.
350,	.98777	.01510	24201.	.96542	.01409	25869.
400.	.98573	.01713	24629.	.96356	.01599	26284.
450.	.98372	.01913	24946.	.96172	.01787	26576.
500.	.98174	.02111	25271.	.95992	.01971	27180.
550.	.97979	.02305	25735.	.95813	.02154	27359.
600.	.97786	.02498	25943.	.95637	.02333	27857.
650.	.97596	.02687	26430.	.95464	.02510	28223.
700.	.97408	.02875	26648.	.95293	.02685	28606.
750.	.97223	.03059	27159.	.95124	.02857	29006.
800.	.97041	.03241	27538.	.94957	.03028	29253.
850.	.96861	.03420	27926.	.94793	.03195	29869.
900.	.96683	.03598	28164.	.94630	.03362	29965.
950.	.96508	.03772	28732.	.94470	.03525	30627.
1000.	.96335	.03945	28982.	.94312	.03687	30939.
1500.	.94723	.05552	31117.	.92838	.05192	33225.
2000.	.93262	.07009	34326.	.91534	.06585	35899.
3000.	.90916	.09348	42757.	.89281	.08824	44666.
4000.	.88991	.11267	52105.	.87475	.10669	54210.
5000.	.87367	.12887	61747.	.85945	.12231	64020.
6000.	.85962	.14287	71407.	.84619	.13585	73882.
7000.	.84728	.15518	81245.	.83452	.14777	83890.
8000.	.83637	.16606	91939.	.82418	.15833	94697.
9000.	.82671	.17569	103856.	.81501	.16769	106798.
10000.	.81815	.18422	117180.	.80688	.17600	120304.
15000.	.7843	.21798	148123.	.7745	.20898	151596.
20000.	.7581	.24410	191427.	.7495	.23457	195422.
30000.	.7208	.28129	268879.	.7137	.27113	273538.
40000.	.6941	.30791	375601.	.6880	.29739	380852.
50000.	.6733	.32865	482076.	.6679	.31789	487749.
60000.	.6562	.34570	586372.	.6514	.33478	592184.
70000.	.6416	.36026	686736.	.6372	.34921	692926.
80000.	.6289	.37292	789591.	.6249	.36178	795818.
90000.	.6176	.38419	887154.	.6140	.37297	893784.
100000.	.6073	.39446	973509.	.6040	.38317	980221.
200000.	.5362	.46536	1410521.	.5349	.45372	1417361.
400000.	.4530	.54831	2410963.	.4540	.53634	2420852.
800000.	.3755	.62559	5176015.	.3783	.61366	5173127.

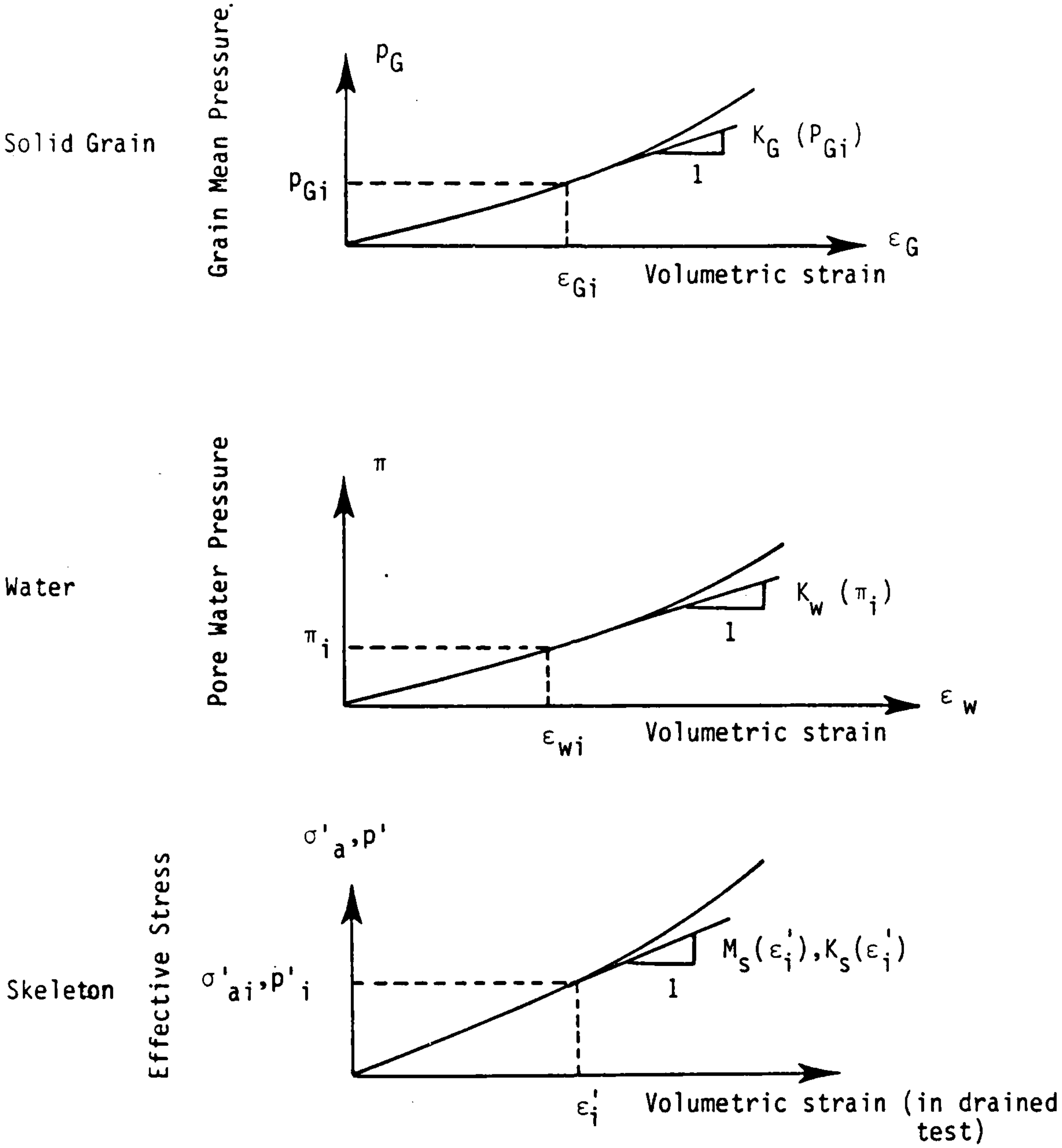


Figure 5.1. Individual component stress-strain curves.

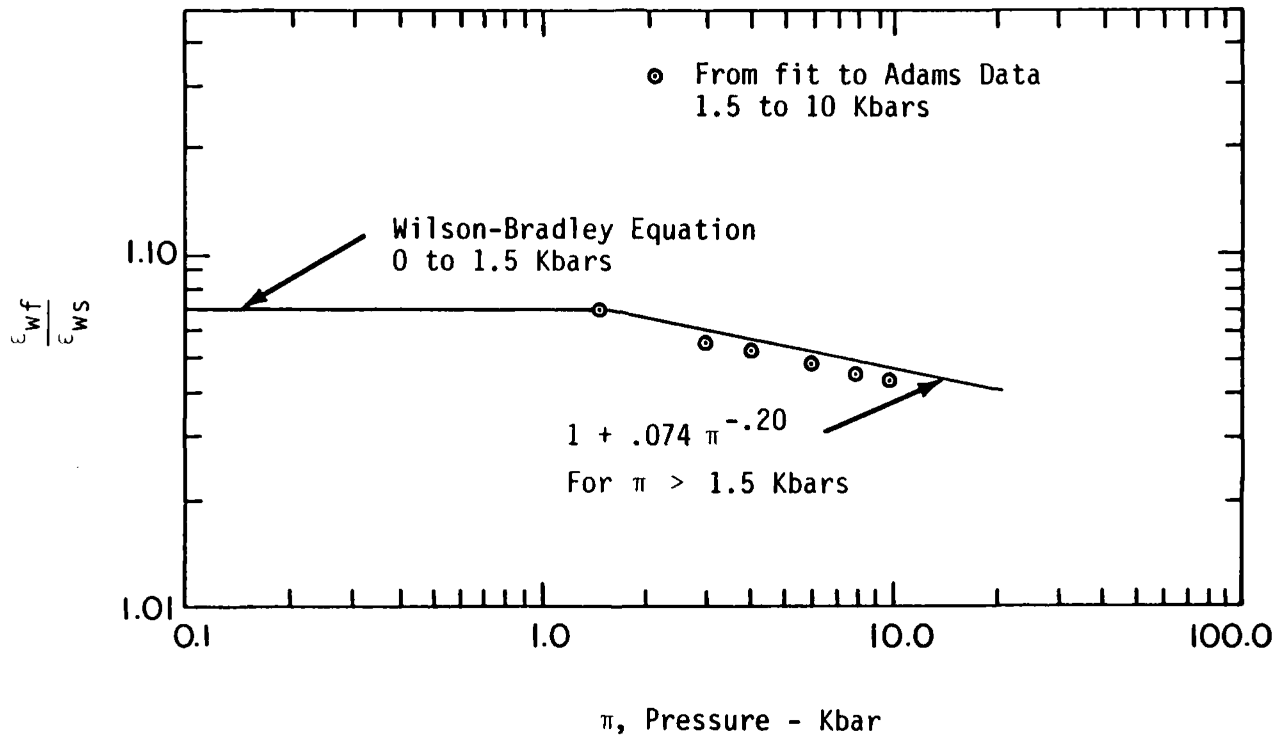


Figure 5.2. Volume strain ratio between fresh water and sea water as a function of pressure.

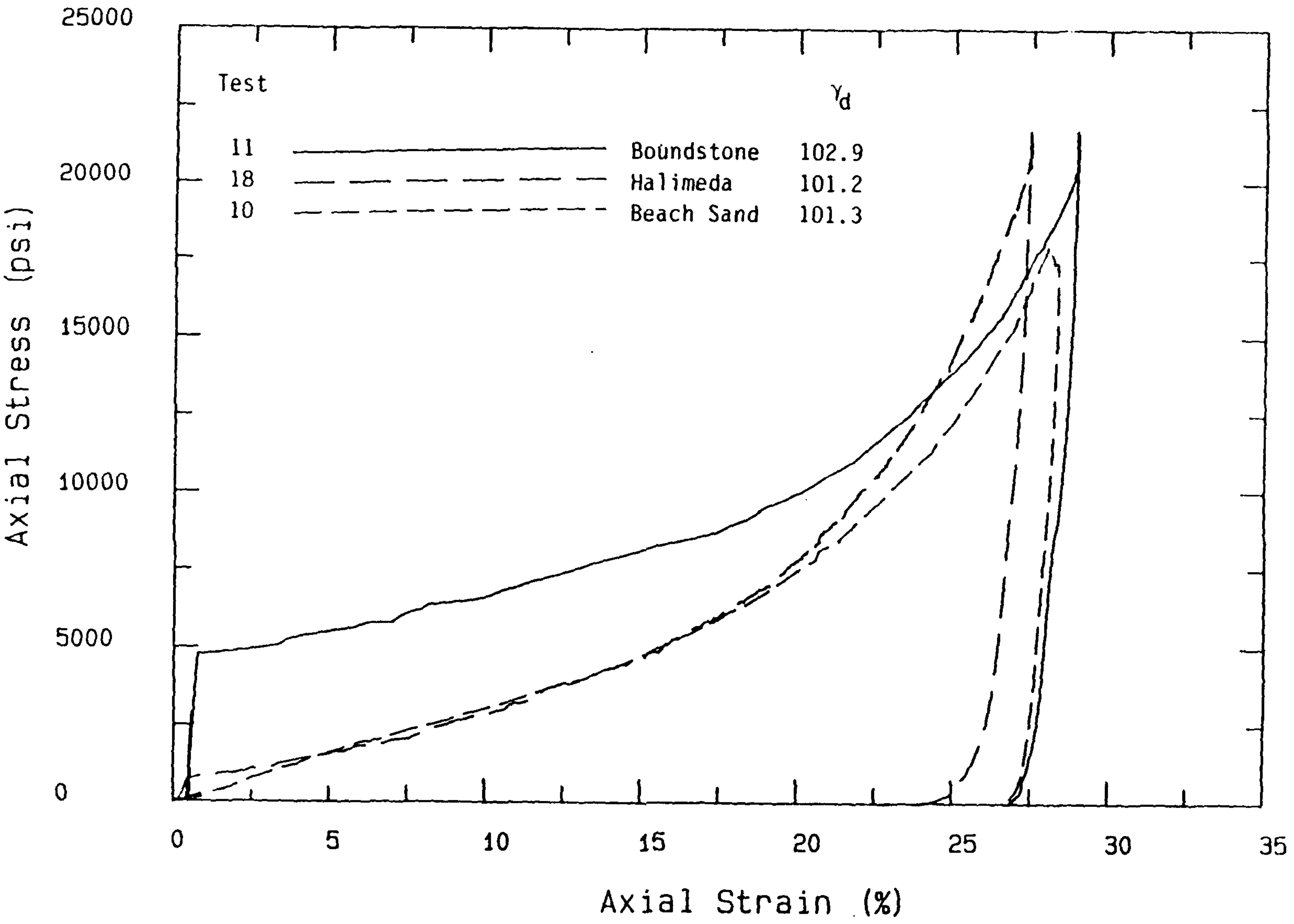


Figure 5.3. Comparison of drained uniaxial stress-strain curves for various carbonate materials.

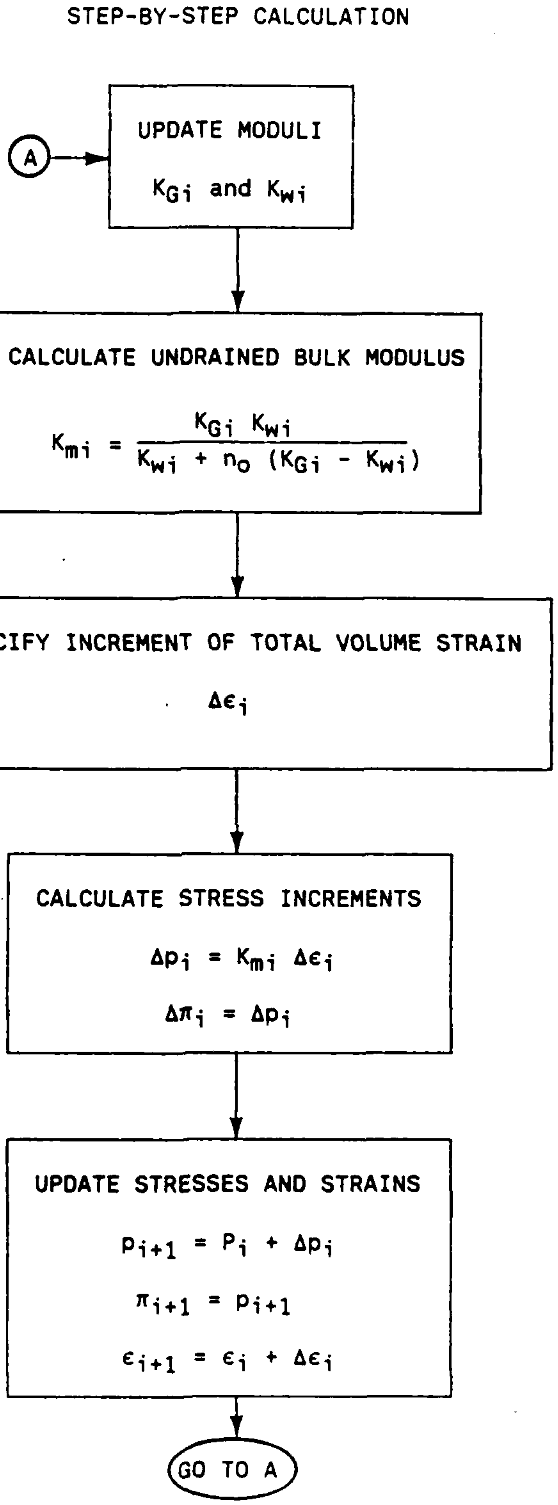

STEP-BY-STEP CALCULATION
A
UPDATE MODULI
K_{Gi} and K_{wi}
CALCULATE UNDRAINED BULK MODULUS
$K_{mi} = \frac{K_{Gi} \; K_{wi}}{K_{wi} + n_o \; (K_{Gi} - K_{wi})}$
SPECIFY INCREMENT OF TOTAL VOLUME STRAIN
$\Delta\epsilon_i$
CALCULATE STRESS INCREMENTS
$\Delta p_i = K_{mi} \; \Delta\epsilon_i$
$\Delta\pi_i = \Delta p_i$
UPDATE STRESSES AND STRAINS
$p_{i+1} = P_i + \Delta p_i$
$\pi_{i+1} = p_{i+1}$
$\epsilon_{i+1} = \epsilon_i + \Delta\epsilon_i$
GO TO A

Figure 5.4. Flow chart for undrained response using mixture model based on engineering strain.

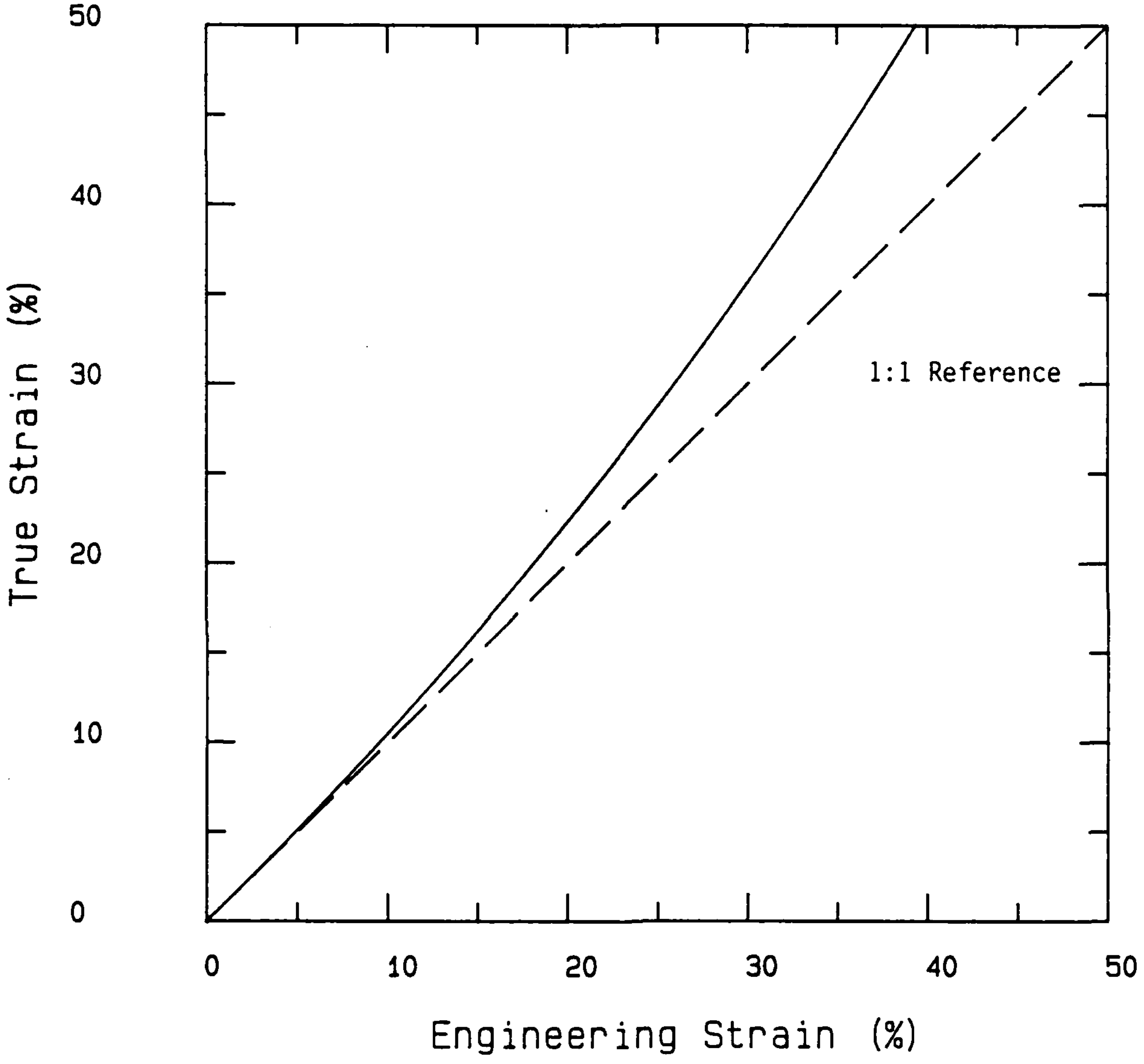


Figure 5.5. Relationship between true strain and engineering strain.

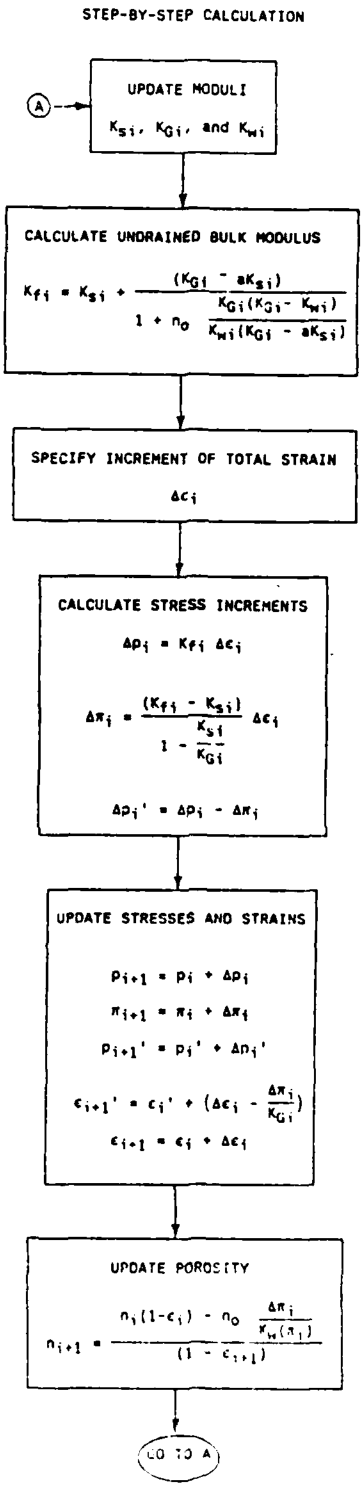
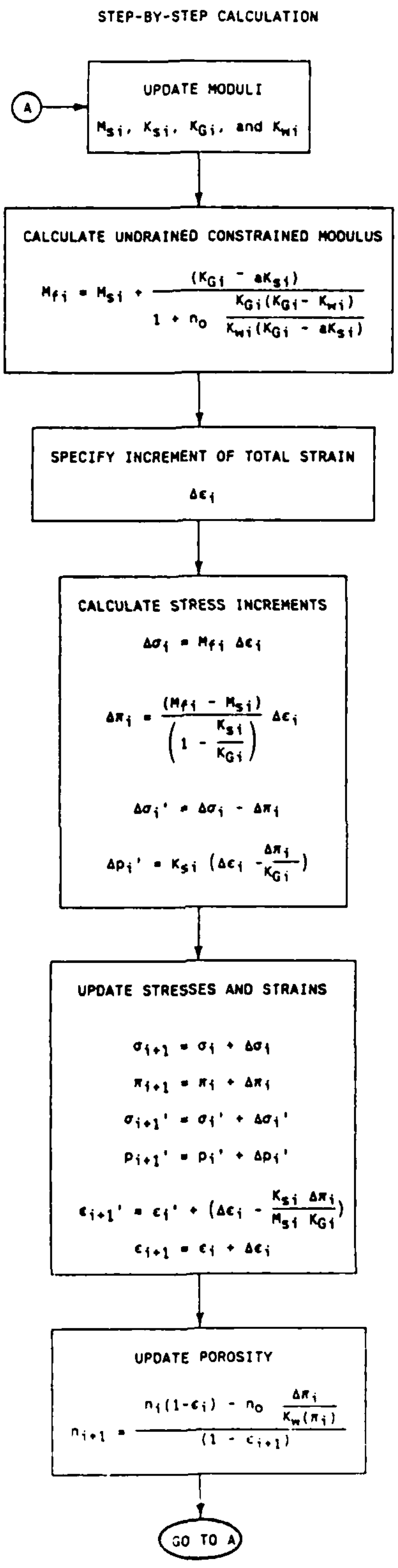

STEP-BY-STEP CALCULATION
A
UPDATE MODULI
K_{si}, K_{Gi}, and K_{wi}
CALCULATE UNDRAINED BULK MODULUS
$K_{fi} = K_{si} + \frac{(K_{Gi} - aK_{si})}{1 + n_o \frac{K_{Gi}(K_{Gi} - K_{wi})}{K_{wi}(K_{Gi} - aK_{si})}}$
SPECIFY INCREMENT OF TOTAL STRAIN
$\Delta\epsilon_i$
CALCULATE STRESS INCREMENTS
$\Delta p_i = K_{fi}\,\Delta\epsilon_i$
$\Delta\pi_i = \frac{(K_{fi} - K_{si})}{1 - \frac{K_{si}}{K_{Gi}}}\,\Delta\epsilon_i$
$\Delta p_i' = \Delta p_i - \Delta\pi_i$
UPDATE STRESSES AND STRAINS
$p_{i+1} = p_i + \Delta p_i$
$\pi_{i+1} = \pi_i + \Delta\pi_i$
$p_{i+1}' = p_i' + \Delta p_i'$
$\epsilon_{i+1}' = \epsilon_i' + (\Delta\epsilon_i - \frac{\Delta\pi_i}{K_{Gi}})$
$\epsilon_{i+1} = \epsilon_i + \Delta\epsilon_i$
UPDATE POROSITY
$n_{i+1} = \frac{n_i(1-\epsilon_i) - n_o \frac{\Delta\pi_i}{K_w(\pi_i)}}{(1 - \epsilon_{i+1})}$
GO TO A

Figure 5.6. Flow chart for undrained bulk response using fully coupled model based on engineering strain.



STEP-BY-STEP CALCULATION
A
UPDATE MODULI
M_{si}, K_{si}, K_{Gi}, and K_{wi}
CALCULATE UNDRAINED CONSTRAINED MODULUS
$M_{fi} = M_{si} + \frac{(K_{Gi} - aK_{si})}{1 + n_o \frac{K_{Gi}(K_{Gi} - K_{wi})}{K_{wi}(K_{Gi} - aK_{si})}}$
SPECIFY INCREMENT OF TOTAL STRAIN
$\Delta\epsilon_i$
CALCULATE STRESS INCREMENTS
$\Delta\sigma_i = M_{fi}\,\Delta\epsilon_i$
$\Delta\pi_i = \frac{(M_{fi} - M_{si})}{\left(1 - \frac{K_{si}}{K_{Gi}}\right)}\Delta\epsilon_i$
$\Delta\sigma_i' = \Delta\sigma_i - \Delta\pi_i$
$\Delta p_i' = K_{si}\,(\Delta\epsilon_i - \frac{\Delta\pi_i}{K_{Gi}})$
UPDATE STRESSES AND STRAINS
$\sigma_{i+1} = \sigma_i + \Delta\sigma_i$
$\pi_{i+1} = \pi_i + \Delta\pi_i$
$\sigma_{i+1}' = \sigma_i' + \Delta\sigma_i'$
$p_{i+1}' = p_i' + \Delta p_i'$
$\epsilon_{i+1}' = \epsilon_i' + (\Delta\epsilon_i - \frac{K_{si}\,\Delta\pi_i}{M_{si}\,K_{Gi}})$
$\epsilon_{i+1} = \epsilon_i + \Delta\epsilon_i$
UPDATE POROSITY
$n_{i+1} = \frac{n_i(1-\epsilon_i) - n_o \frac{\Delta\pi_i}{K_w(\pi_i)}}{(1 - \epsilon_{i+1})}$
GO TO A

Figure 5.7. Flow chart for undrained uniaxial strain response using fully coupled model based on engineering strain.

SECTION 6

UPDATED TWO-PHASE MODEL AND COMPUTATIONAL ALGORITHM

6.1 INTRODUCTION

The development of advanced multiphase models and efficient computational algorithms for handling multiphase models is moving ahead rapidly. The ultimate goal is the incorporation of an advanced three-phase model into an efficient algorithm for treatment of a broad spectrum of multiphase dynamic problems. Because this is a multi-year effort, we felt it was essential to incorporate many of these advances into an interim two-phase code. The purposes of this interim approach are:

- To evaluate advances in two-phase material models;
- To evaluate efficient calculational algorithms and techniques which will be incorporated into the multiphase code;
- To provide a means of calculating and studying two-phase phenomenology including wave propagation problems, influence of material and flow parameters on the response of two-phase porous media, etc.;
- To provide a means of calculating and designing both laborotory and field tests;
- To provide an independent means of verification of the multiphase code MPDAP when it becomes available.

This section describes the advances in material models and code development that have been incorporated into the interim two phase dynamic analysis program, TPDAPII. TPDAPII is a vastly improved evolutionary follow-on to our original two-phase code, TPDAP, documented by Kim and Blouin, 1984. The advanced material models and computational techniques included in TPDAPII are presented in this section. Section 6.2 describes three different material model options including the uniaxial strain model, the decoupled elasto-plastic model and the ARA2D plasticity model. Section 6.3 describes a new single-point stress calculational technique for computing stresses, strains, and nonlinear constitutive properties. Section 6.4 includes a series of verification problems which exercise both single phase and two-phase options of the code in static, quasi-static and dynamic problems with known solutions.

An essential part of the TPDAPII development was the preparation of the User's Manual included in Appendix G. A complete listing of TPDAPII is included in Appendix H.

6.2 MATERIAL MODEL DESCRIPTIONS

6.2.1 Uniaxial Strain Model (UNIAX)

The uniaxial strain model is used for one dimensional plane strain loadings only. The constrained loading modulus of the material skeleton is specified in incremental nonlinear form as a function of the applied vertical effective stress. Thus, laboratory drained uniaxial loading data can be input to the code.

There are three options available for use in unloading. In the first option, the unloading constrained modulus is specified incrementally as a function of current effective vertical stress. In this option a multilinear unloading curve can be input. The second option uses a constant unloading modulus which is dependent on the previous maximum vertical effective stress. The third option also uses a constant unloading modulus which is expressed as an exponential function of the previous maximum vertical effective stress, σ'_{vmax}. The unloading modulus, M_u, is expressed as

$$M_u = c(\sigma'_{vmax})^d \qquad (6\text{-}1)$$

where c and d are experimentally determined constants. Blouin et al. (1984) determined values of c and d of 3470 and 0.651 respectively for Enewetak beach sand, with both M_u and σ'_{vmax} expressed in psi. Details on both the loading and unloading routines are provided on cards 15 and 17 of the user's manual in Appendix E.

In the uniaxial strain model and all other models, whenever the effective vertical stress drops to zero, liquefaction has occurred and the skeleton unloading modulus is automatically set to zero.

6.2.2 Decoupled Elasto-plastic Model (DCOUP)

The decoupled elasto-plastic model is the simplest model for use in general multidimensional loadings. Since it is a decoupled model, the deviatoric and volumetric behavior are independent of one another. Thus, at failure the material does not exhibit dilatency (increase in volume).

The volumetric behavior, as shown schematically on card 16B of the user's manual, is described by loading and unloading bulk skeleton moduli expressed as a function of effective mean pressure. Thus, the volumetric model can be tailored to the results of isotropic drained test data.

The deviatoric behavior is illustrated in Figures 6.1 through 6.3. As shown in Figure 6.1, the shear stress lies in three distinct zones which are a function of the mean effective stress, p'. Below the initial yield surface the material response is linear-elastic. The shear modulus, G, in this zone is constant with

$$G = G_e \qquad (6\text{-}2)$$

where G_e is the elastic shear modulus of the skeleton. Above the initial yield surface and below the yield surface, the material transitions from the elastic to the plastic state. In this zone, as shown in Figure 6.3, the shear modulus varies linearly between G_e and the plastic shear modulus, G_p, and is given by

$$G = (G_e - G_p)\,\frac{(1-\eta)}{(1-a)} + G_p \qquad (6\text{-}3)$$

where

$\eta = \dfrac{\tau}{\tau_y}$ and

a defines the initial yield strength as a fraction of the yield stress

τ is the current octahedral shear stress and

τ_y is the octahedral shear strength specified as a function of mean effective stress

The octahedral shear stress, τ, is given by

$$\tau = \frac{\sqrt{2}}{3} (\sigma'_a - \sigma'_r) \qquad (6\text{-}4)$$

where

σ'_a is the axial effective stress and

σ'_r is the radial effective stress

Above the yield surface and below the failure surface the shear modulus is again constant with

$$G = G_p \qquad (6\text{-}5)$$

where G_p is the plastic shear modulus. As shown in Figure 6.2, above the failure surface the shear modulus is given by

$$G = bG_p \qquad (6\text{-}6)$$

where b is a user defined fraction of the plastic shear modulus.

As shown in Figure 6.2, the unloading modulus is assumed to be constant with a value of G_e. Unload-reload cycles are governed by the previous maximum shear stress. If the maximum previous shear stress is greater than $a\tau_y$, as shown in Figure 6.1, a new initial yield condition is established based on the principal of isotropic hardening, with $a\tau_y$ redefined as equal to the maximum previous shear stress. For all subsequent values of shear stress less than the newly defined $a\tau_y$, both the loading and unloading shear moduli are elastic and equal to G_e.

6.2.3 ARA Two-Dimensional Elasto-plastic Model (ARA2D)

The most sophisticated material model in the current version of TPDAPII is the elasto-plastic ARA two dimensional model, ARA2D (Kim, Piepenburg and Merkle, 1986). This is an elastic-perfectly plastic coupled model with the volumetric and deviatoric behaviors dependent upon one another once the failure surface is reached. The failure surface is defined by

$$F(p,q,\theta) = q - \left[(\alpha + \beta p)^n + \kappa\right] R(\theta) = 0 \qquad (6\text{-}7)$$

where

the failure surface, F, is a function of the mean normal stress, p, the shear stress, q, and the load angle, θ.

These stress invariants are defined as follows:

$$p = 1/3\ (\sigma_1 + \sigma_2 + \sigma_3) \qquad (6\text{-}8)$$

$$q = \sqrt{3J_2} \qquad (6\text{-}9)$$

$$\theta = 1/3\ \sin^{-1}\left(-\frac{27}{2}\frac{J_3}{q^3}\right) \qquad (6\text{-}10)$$

where:

σ_1, σ_2, and σ_3 are the principal stresses, and J_2 and J_3 are the second and third deviatoric stress invariants, respectively.

The function R(θ) describes the shape of the yield surface, as projected in the π plane (octahedral plane). Figure 6.4 shows the influence of the parameter K on the shape of the yield surface. K is the ratio of the shear strength in triaxial extension to the shear strength in triaxial compression at the same mean pressure. K is a measure of the influence of the intermediate principal stress on the yield surface and can vary from 0.5 to 1.0. When K is equal to unity R(θ) is circular, indicating a Drucker-Prager or Von Mises failure model. When K is less than unity, R(θ) is a smooth cornered approximation to the Mohr-Coulomb failure envelope.

The parameter n in Equation 6-7 determines the shape of the yield surface in the p-q plane. For n = 0, the shear strength is constant with respect to the mean pressure and the strength envelope reduces to the Von Mises or Tresca yield surface. For n = ½, the strength envelope represents Hoek and Brown's (1982) failure surface. This nonlinear failure model is a multidimensional generalization of the original one-dimensional axisymmetric Hoek and Brown model which is based on extensive laboratory and field data (Kim, Piepenburg and Merkle, 1986). For n = 1, shear strength is linearly proportional to the mean pressure and the strength envelope in the p-q plane is representative of the Drucker-Prager or Mohr-Coulomb failure surface.

The parameters α, β and κ of Equation 6-7 define the failure envelope in the p-q plane. They can be determined from laboratory tests. Recommended relationships for determining these parameters for Von Mises, Hoek and Brown and Mohr-Coulomb type materials are listed in Tables 6.1 and 6.2.

6.3 A SINGLE-POINT METHOD OF COMPUTING STRAINS, STRESSES, AND CONSTITUTIVE RELATIONSHIPS

6.3.1 Introduction

In the conventional method used for nonlinear finite element analysis, the integration points (i.e. Gauss points) represent not only the sampling points used to integrate the element stiffness matrix and element residual load vector, but also the locations where the strains, stresses, and constitutive relationships are updated. For large scale nonlinear structural or geotechnical analysis this conventional method requires an unduly large storage capacity to save historical element data and considerably longer computational time. Furthermore, for nonlinear spherically or axisymmetrically divergent problems, this conventional method may develop local stress oscillations which cause numerical instabilities.

The concept of the proposed single point method was developed under DNA contract by Kim, Piepenburg and Merkle (1986). This new method uses the same integration points to integrate the element stiffness matrix and element residual load vector as the conventional method. The strains, stresses, and constitutive equations, however, are evaluated at the center of each individual element instead of at the integration points (refer to Figure 6.5). This new single-point scheme eliminates local stress oscillations for geometrically divergent problems and significantly reduces data storage requirements and computational time. For clarification, the single point method for nonlinear static analysis is described below. A more complicated formulation for two phase media has been incorporated into TPDAPII.

6.3.2 Description of the Single Point Method

The usual procedure for solving nonlinear structural equilibrium equations in static finite element problems uses the incremental step-by-step. approach;

$$[K_T]_{n-1} \{\Delta\bar{U}\}_n = \{P\}_n - \{R\}_{n-1} \qquad (6\text{-}11)$$

where

$\{\Delta\bar{U}\}_n$ is the structural nodal displacement increment vector at step n;

$\{P\}_n$ is the structural applied nodal force vector at step n.

The structural tangent stiffness matrix, $[K_T]_{n-1}$ is expressed as

$$[K_T]_{n-1} = \Sigma\int_V [B]^T \; [D^{ep}_{n-1}][B]dv \qquad (6\text{-}12)$$

where

$[B]$ is the matrix which relates the change in the strain field, $\{\Delta\epsilon\}$, to the element nodal displacement increment vector $\{\Delta\bar{u}\}$;

$[D^{ep}_{n-1}]$ is the constitutive elasto-plastic matrix which relates incremental stresses to incremental strains;

$[B]^T$ is the transpose of $[B]$ and v denotes volume integration.

The structural residual load vector, $\{R\}_{n-1}$, is given by

$$\{R\}_{n-1} = \Sigma \int_V [B]^T \{\sigma_{n-1}\} dv \tag{6-13}$$

where $\{\sigma_{n-1}\}$is the stress field at step n-1 and Σ in Equations 6-12 and 6-13 indicates the contribution of all elements.

The single point method assumes that the constitutive matrix, $[D^{ep}_{n-1}]$, is constant within each individual element and is evaluated at the center of each element, thus

$$[D^{ep}] = [D^{ep}]_c \tag{6-14}$$

where subscript c indicates the evaluation of the matrix at the element center. Substituting Equation 6-14 into 6-12 yields

$$\boxed{[K_T]_{n-1} = \Sigma \int_V [B]^T \; [D^{ep}_{n-1}]_c \; [B] dv} \tag{6-15}$$

Thus the strains, stresses and other historical element data are required to be updated only at the center of each element.

In order to be consistent with the above assumption, the structural residual load vector should be altered as follows:

$$\{R\}_{n-1} = \{R\}_{n-2} + \{\Delta R\}_{n-1} \tag{6-16}$$

where

$$\{\Delta R\}_{n-1} = \Sigma \left[\int_V [B]^T [D^{ep}_{n-2}] c [B] dv \right] \{\Delta \bar{u}\}_{n-1} \tag{6-17}$$

As a matter of practicality, a simpler alternate approach was used in TPDAPII. The residual load vector of Equation 6-13 can be approximated by substitution of the resultant stress field at the element center, $\{\sigma_{n-1}\}_c$, as

$$\boxed{\{R\}_{n-1} = \Sigma\int_V [B]^T \{\sigma_{n-1}\}_c \; dv} \qquad (6\text{-}18)$$

Equations 6-15 and 6-18 have been implemented in TPDAPII, and the validation of this new single point technique is demonstrated in verification problems 1 and 3 in subsections 6.4.2 and 6.4.4.

6.4 VERIFICATION PROBLEMS

6.4.1 Introduction

The objective of this section.is to demonstrate the validity of the computer code, TPDAPII. Five verification problems for which analytical solutions are available are presented here. Table 6.3 summarizes these verification problems in terms of analysis type, geometric model, loading condition, and material model.

6.4.2 Verification Problem 1: Elastic Spherical Wave Propagation in One-Phase Medium

Figure 6.6 shows a 12 inch hollow spherical hole in an infinite elastic medium subjected to a 100 psi internal step load. Material properties and time-steps used for the calculations are included in Figure 6.6. Two calculations were performed to demonstrate the efficiency and accuracy of the new single-point technique (Section 6.3) for computing stresses and strains. The first calculation uses this single-point technique to compute stresses and strains at the center of individual elements. The second calculation uses the conventional method to compute stresses and strains at the integration points of individual elements.

Results from both calculations are compared in Figures 6.7 through 6.9. Figure 6.7 shows the profiles of radial displacement at 55 msec. Both methods give good agreement with the closed form solution. Radial and tangential stress profiles at 65 msec are plotted in Figures 6.8 and 6.9, respectively. Both methods smear the stresses around the shock wave front. This smearing could be minimized by reducing the element sizes. Near the spherical hole, the stresses generated by the conventional method are oscillating within individual elements while the stresses generated using the single-point technique give an excellent approximation of the closed form solution.

6.4.3 Verification Problem 2: Elastic Spherical Wave Propagation in One-Phase Medium Using Skew Boundary

The same problem as presented in Section 6.4.2 is solved by 2-D axisymmetric analysis with a skew boundary. As shown in Figure 6.10 the skew boundary is oriented such that it allows motion only in the radial direction in order to model spherical wave propagation.

Figures 6.11 and 6.12 show the radial and tangential stress profiles at 55 msec, respectively, calculated by the single-point technique for computing stresses and strains. The stress profiles are essentially the same as those calculated using the single-point technique with spherical symmetry presented in verification problem 1.

6.4.4 Verification Problem 3: Static Nonlinear Tunnel Analysis in One-Phase Medium

Figure 6.13 shows a schematic section view of a 64 inch diameter circular tunnel subjected to a uniform free-field loading of 2300 psi. The material is represented by the ARA2D model and is assumed to be elastic under the failure surface and perfectly plastic along the failure surface. The Drucker-Prager model ($n = 1$) corresponding to the triaxial compression mode ($K = 1$), is used to represent the elasto-plastic rock. The elastic and plastic properties of the of the rock are given in Figure 6.13.

As in verification problem 1, two calculations using the new single-point method and the conventional method are performed to compute stresses, strains and nonlinear constitutive relationships in individual elements. In Figure 6.14 the radial stress profiles from both calculations are compared with the analytical finite difference solution presented by Kim, Davister and Piepenburg (1986). As was the case in the spherical wave propagation problem, the conventional method shows stress oscillations within each element while the single-phase method provides an excellent match with the finite difference solution. It should be noted that such numerical oscillations generated by conventional methods are more pronounced near the tunnel surface and can be the cause of numerical instabilities when the large volumes of rock surrounding the tunnel are subjected to plastic yielding. Oscillations are generally much more pronounced during plastic yielding.

6.4.5 Verification Problem 4: Linear 1-D Strain Consolidation

A fully saturated soil deposit is assumed to overlay a rigid impermeable base. The soil deposit is subjected to a step loading at the ground surface, which is assumed to be a free-draining boundary. The initial excess pore pressure distribution is assumed to be constant throughout the deposit. Twenty equally spaced 4-node elements are used with a non-dimensional time increment factor ΔT = 0.005. Plotted in Figure 6.15 is the profile of the normalized excess pore water pressures at time factor T = 0.5, at which time about 75% of the excess pore pressures is dissipated on average throughout the soil deposit. The calculated excess pore water pressures show close agreement with Terzaghi's exact solution.

6.4.6 Verification Problem 5: 1-D Plane Strain Elastic Wave Propagation Through Two-Phase Medium

A step load is applied to the free surface of a fully saturated porous elastic semi-infinite medium. Figure 6.16 shows the loading time-history and material properties used in the analysis. The porous medium is discretized into 150 equally spaced 4-node elements to a depth of 225 ft. A constant time step of 0.155 msec is used.

Figure 6.17 shows the computed stresses in the soil skeleton and pore water at 31 msec. The scale on the effective stress is magnified 100 times compared to the pore pressure scale. It shows that the pore water pressure is more than a hundred times higher than the effective stress. The stress magnitudes and stress wave locations correspond closely to the hand calculated values computed from Blouin and Kim (1984). Also plotted in the figure is the one-phase total stress analysis using the equivalent undrained modulus. The two-phase stress profiles are considerably more smeared than the corresponding one-phase stress profiles, indicating that the additional smearing at the wave front is due to the fluid damping associated with frictional energy dissipation between the pore water and the porous solid skeleton.

Table 6.1. Material Constants: α, β, and κ in ARA2D Model.

	$n = 0$ Von Mises or Tresca	$n = ½$ Hoek and Brown	$n = 1$ Mohr-Coulomb or Drucker-Prager
α	N/A	$(\frac{m^2}{36} + s)\ \sigma_c{}^2$	1000
β	N/A	$m\sigma_c$	$\frac{6 \sin\phi}{(3-\sin\phi)}$
κ	$q' - 1$	$\frac{1}{6}\ m\sigma_c$	$\frac{3(1-\sin\phi)}{(3-\sin\phi)}\ \sigma_c - 1000$

Note: $q' = \sigma_1 - \sigma_3$

where σ_1 and σ_3 are the major and minor principal stresses at failure and

σ_c = unconfined compressive strength

ϕ = internal friction angle

m,s = Hoek and Brown's material constants as tabulated in Table 6.2

Table 6.2 Empirical Rock Material Parameters in Hoek and Brown's Model

APPROXIMATE EQUATIONS FOR PRINCIPAL STRESS RELATIONSHIPS AND MOHR ENVELOPES FOR INTACT ROCK AND JOINTED ROCK MASSES					
	CARBONATE ROCKS WITH WELL DEVELOPED CRYSTAL CLEAVAGE *dolomite, limestone and marble*	LITHIFIED ARGILLACEOUS ROCKS *mudstone, siltstone, shale and slate (normal to cleavage)*	ARENACEOUS ROCKS WITH STRONG CRYSTALS AND POORLY DEVELOPED CRYSTAL CLEAVAGE *sandstone and quartzite*	FINE GRAINED POLYMINERALLIC IGNEOUS CRYSTALLINE ROCKS *andesite, dolerite, diabase and rhyolite*	COARSE GRAINED POLYMINERALLIC IGNEOUS AND METAMORPHIC CRYSTALLINE ROCKS *amphibolite, gabbro, gneiss, granite, norite and quartz-diorite*
INTACT ROCK SAMPLES *Laboratory size rock specimens free from structural defects* *CSIR rating 100+, NGI rating 500*	$\frac{m = 7}{s = 1}$	$\frac{10}{1}$	$\frac{15}{1}$	$\frac{17}{1}$	$\frac{25}{1}$
VERY GOOD QUALITY ROCK MASS *Tightly interlocking undisturbed rock with unweathered joints spaced at ±3 metres* *CSIR rating 85, NGI rating 100*	$\frac{3.5}{0.1}$	$\frac{5}{0.1}$	$\frac{7.5}{0.1}$	$\frac{8.5}{0.1}$	$\frac{12.5}{1}$
GOOD QUALITY ROCK MASS *Fresh to slightly weathered rock, slightly disturbed with joints spaced at 1 to 3 metres.* *CSIR rating 65, NGI rating 10*	$\frac{0.7}{0.004}$	$\frac{1.0}{0.004}$	$\frac{1.5}{0.004}$	$\frac{1.7}{0.004}$	$\frac{2.5}{0.004}$
FAIR QUALITY ROCK MASS *Several sets of moderately weathered joints spaced at 0.3 to 1 metre.* *CSIR rating 44, NGI rating 1.0*	$\frac{0.14}{0.0001}$	$\frac{0.20}{0.0001}$	$\frac{0.3}{0.0001}$	$\frac{0.34}{0.0001}$	$\frac{0.5}{0.0001}$
POOR QUALITY ROCK MASS *Numerous weathered joints spaced at 30 to 500mm with some gouge filling / clean waste rock* *CSIR rating 23, NGI rating 0.1*	$\frac{0.04}{0.00001}$	$\frac{0.05}{0.00001}$	$\frac{0.08}{0.00001}$	$\frac{0.09}{0.00001}$	$\frac{0.13}{0.00001}$
VERY POOR QUALITY ROCK MASS *Numerous heavily weathered joints spaced less than 50mm with gouge filling / waste rock with fines* *CSIR rating 3, NGI rating 0.01*	$\frac{0.007}{0}$	$\frac{0.01}{0}$	$\frac{0.015}{0}$	$\frac{0.017}{0}$	$\frac{0.0025}{0}$

Table 6.3. Summary of Verification Problems.

Problem Number	Analysis Type	Geometric Model	Loading Condition	Material Model
1	one-phase dynamic	1-D spherical symmetry	heaviside step load	linear elastic
2	one-phase dynamic	2-D axi-symmetric with skew boundary	heaviside step load	linear elastic
3	one-phase static	1-D axi-symmetric	uniform overpressure load	elasto-plastic (ARA2D model)
4	consolidation	1-D plane strain	heaviside step load	linear elastic
5	two-phase dynamic	1-D plane strain	heaviside step load	linear elastic

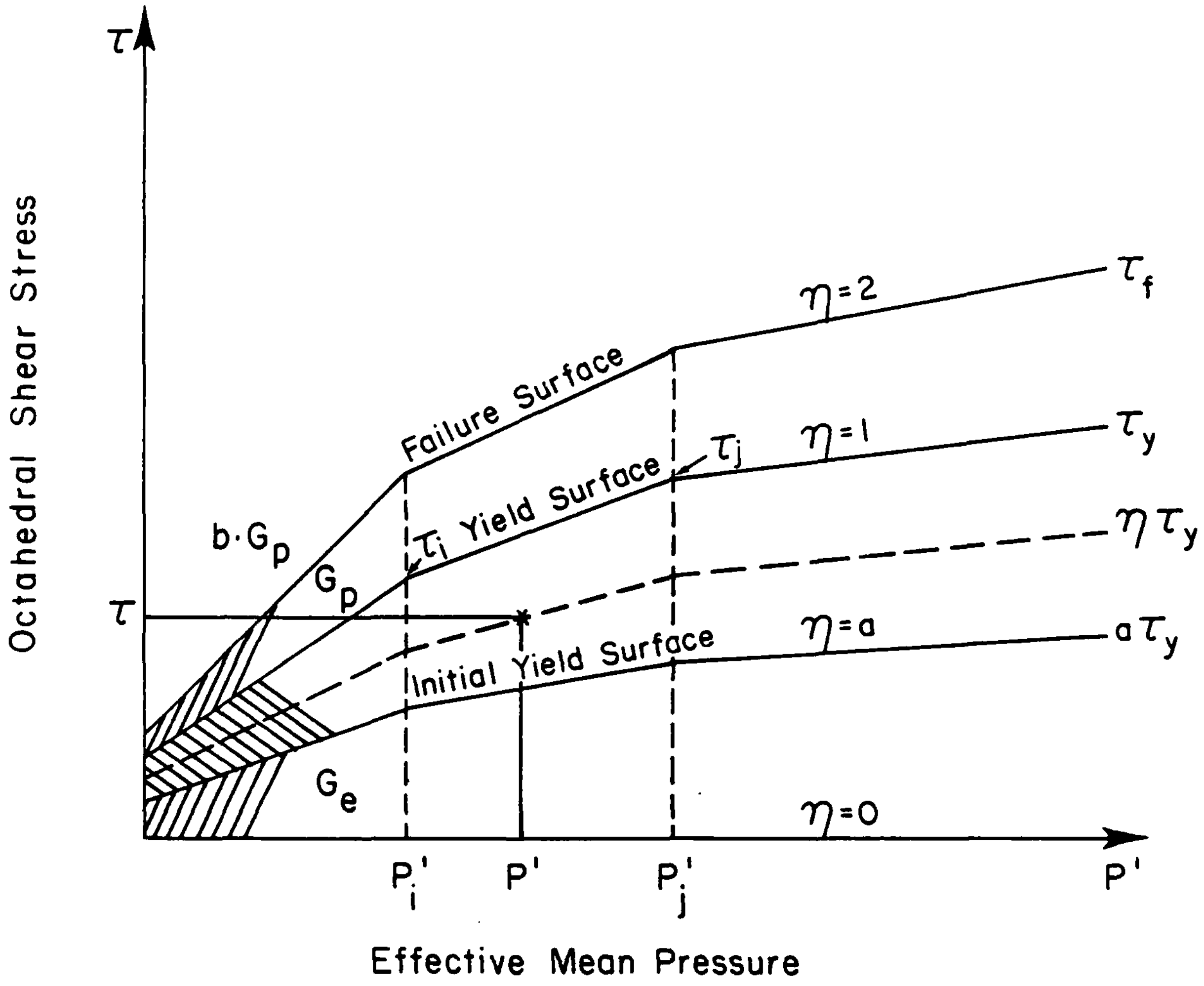


Figure 6.1. Shear strength envelope used in decoupled elasto-plastic model.

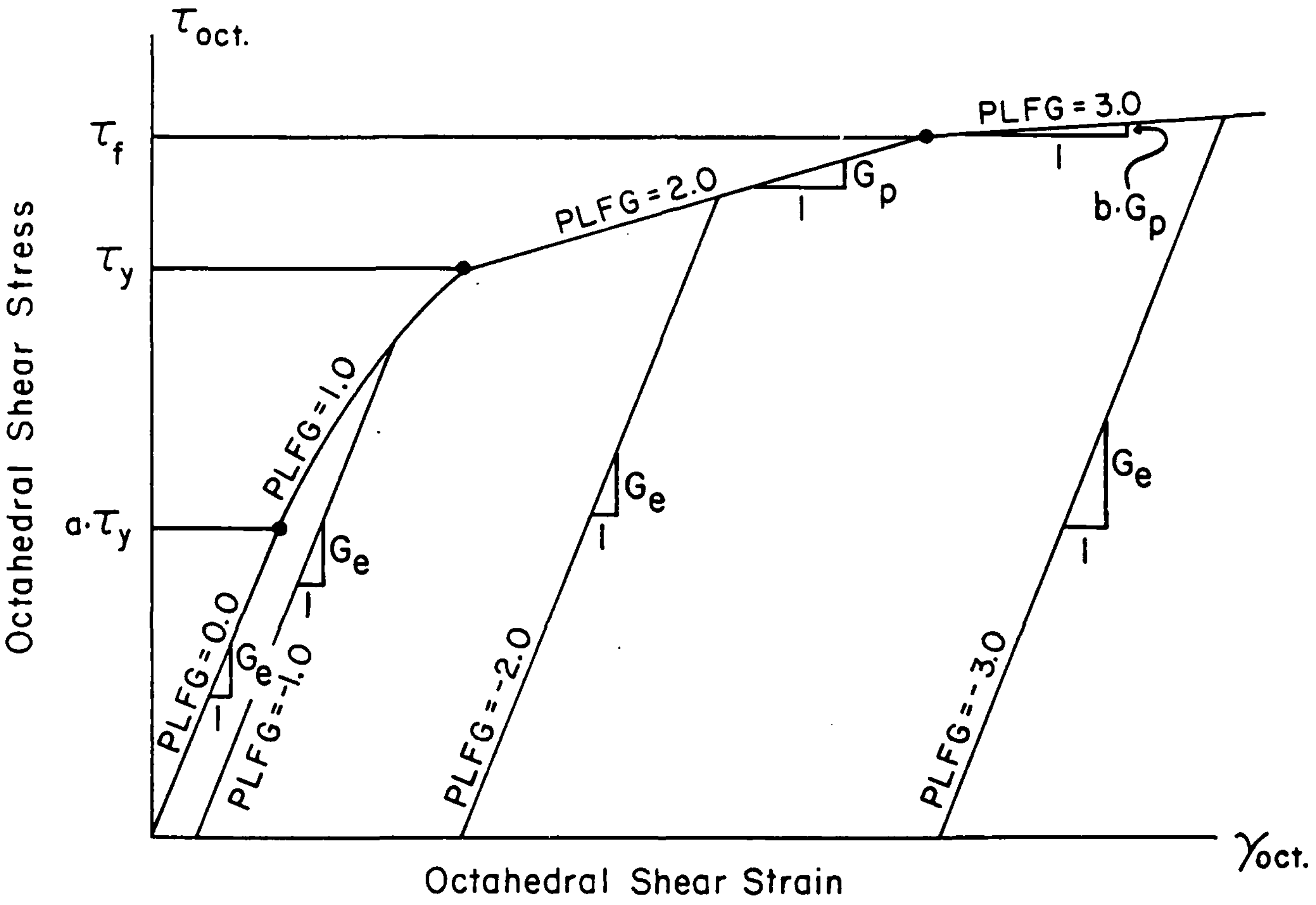


Figure 6.2. Shear stress-strain behavior in decoupled elasto-plastic model.

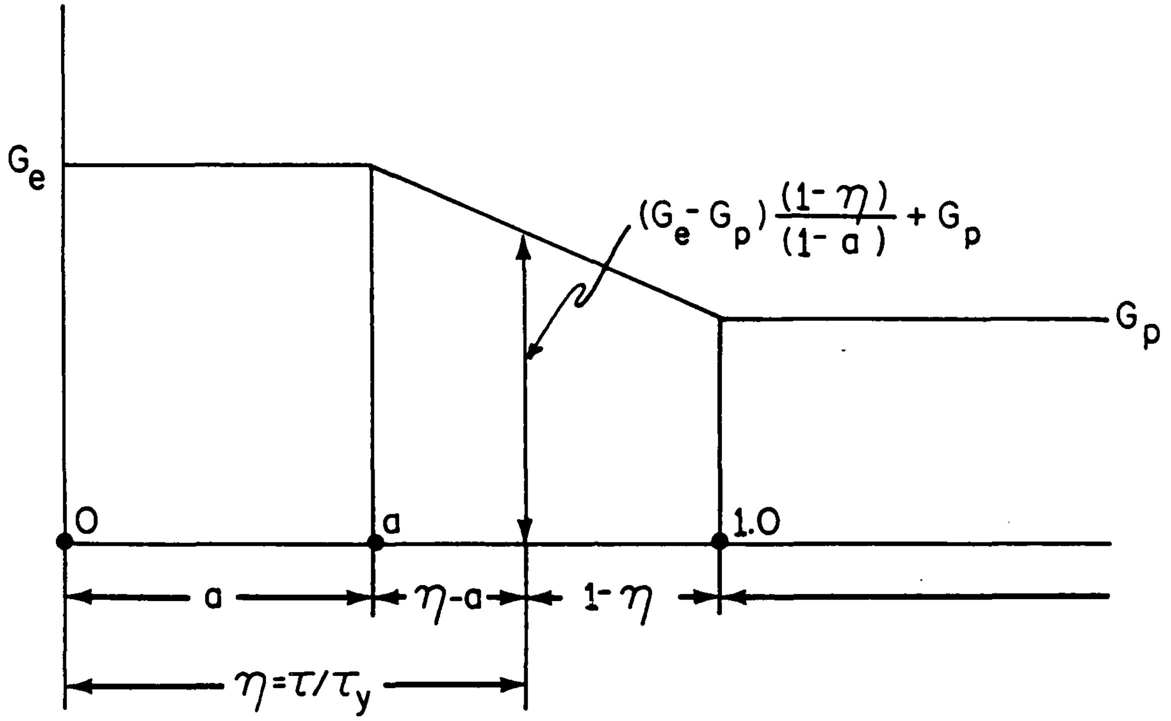


Figure 6.3. Shear modulus in transition state, decoupled elasto-plastic model.

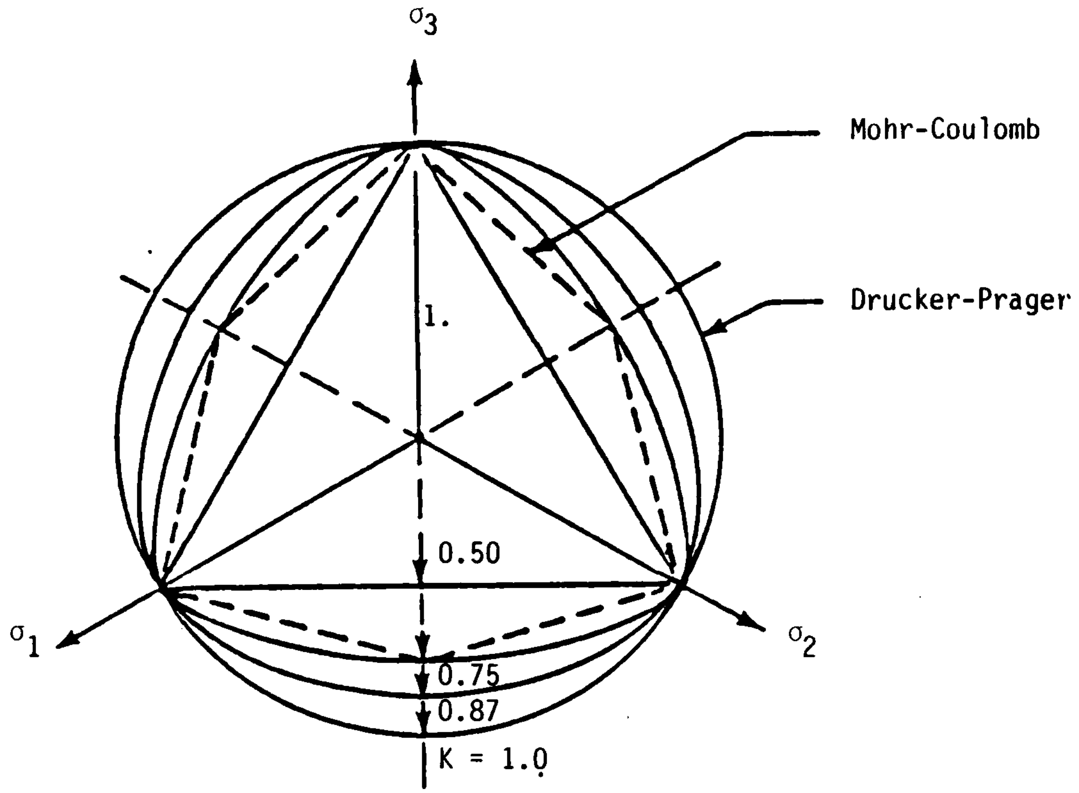
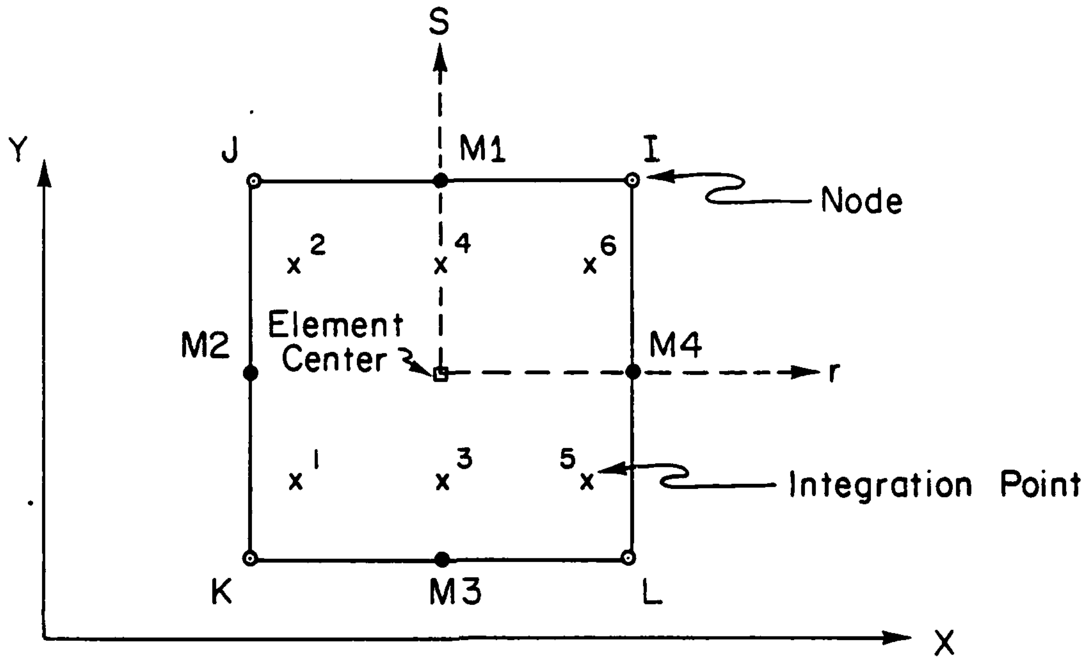
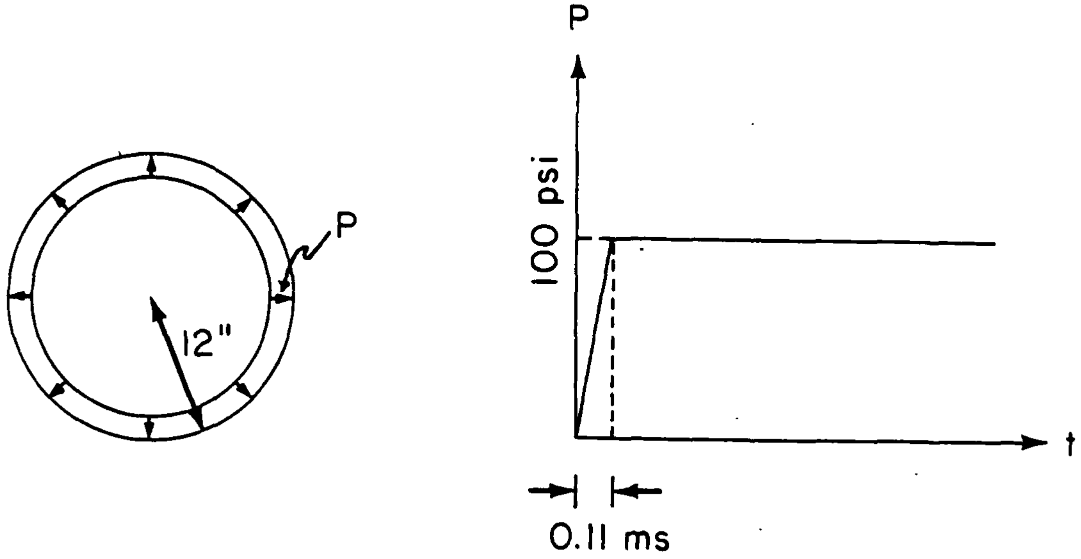


Figure 6.4. Shape of strength envelope, R(θ), on octahedral plane.



Single Element

Figure 6.5. Schematic view to illustrate the use of single-point technique.



Time Step	Δt = 0.022 Msec
Young's Modulus	E = 12,457 psi
Poisson's Ratio	ν = 0.25
Mass Density	$\rho = 1.88 \times 10^{-4}$ lb-sec^2/in^4

Figure 6.6. Verification problem 1, elastic spherical wave propagation in one-phase medium.

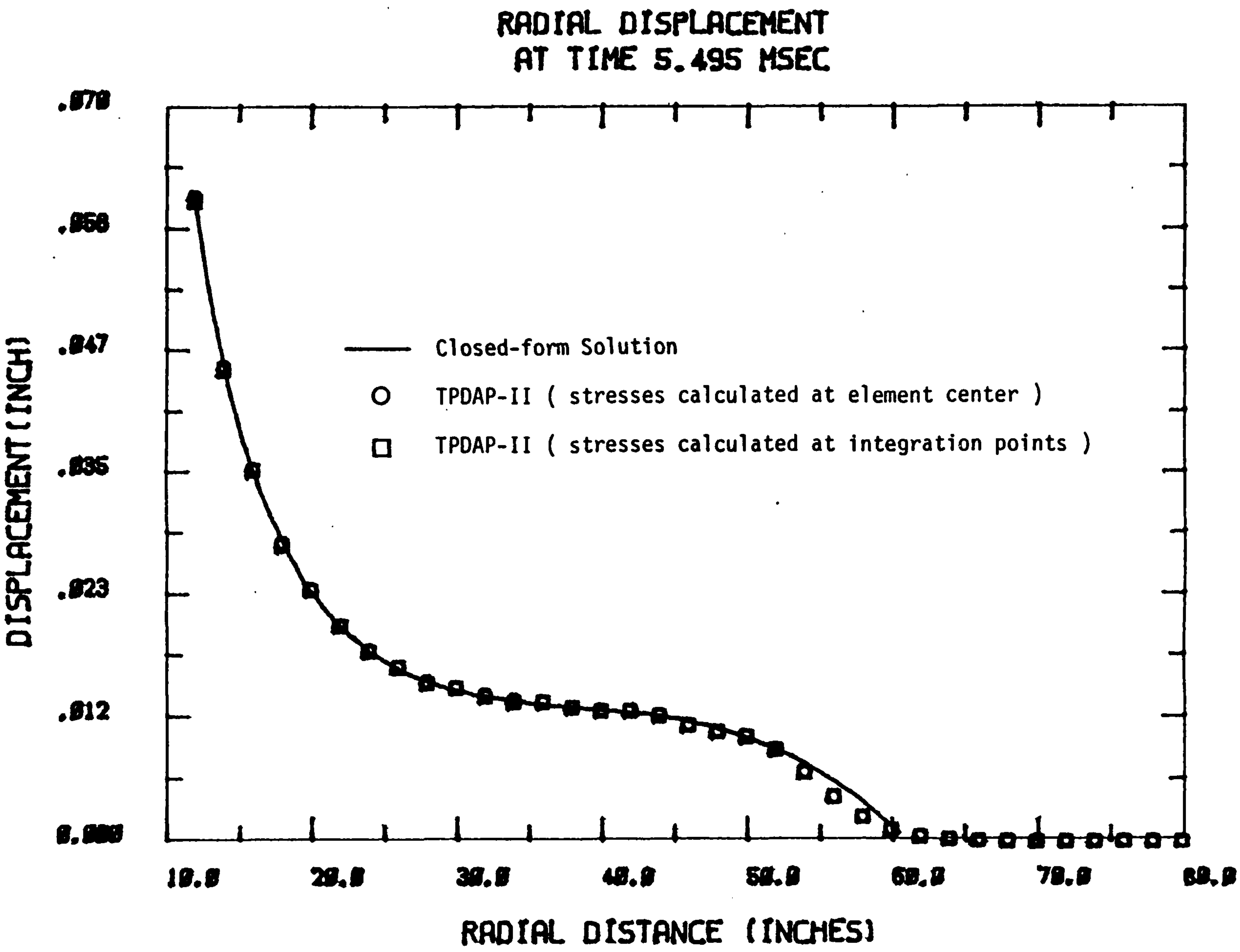


Figure 6.7. Radial displacement profile, verification problem 1.

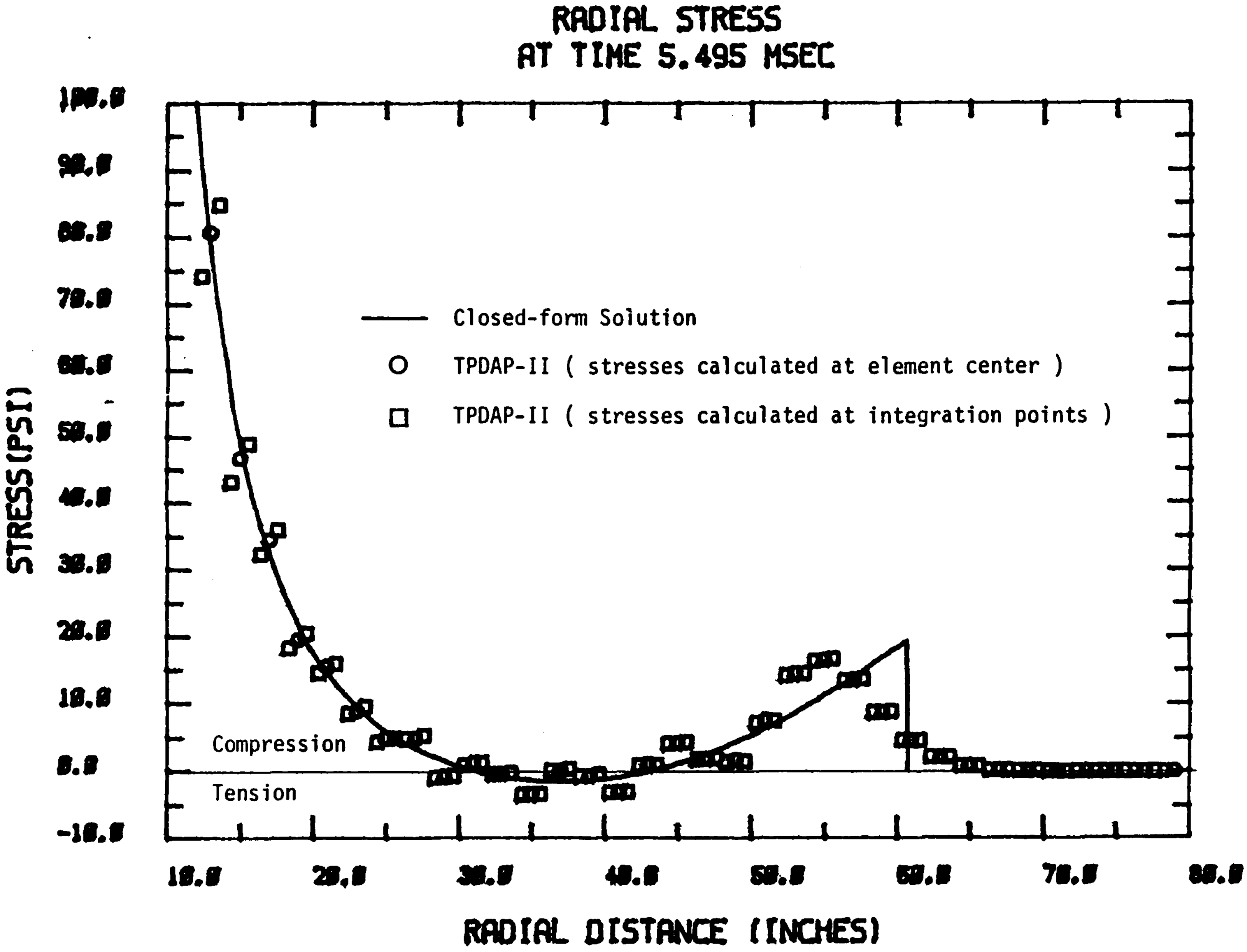


Figure 6.8. Radial stress profile, verification problem 1.

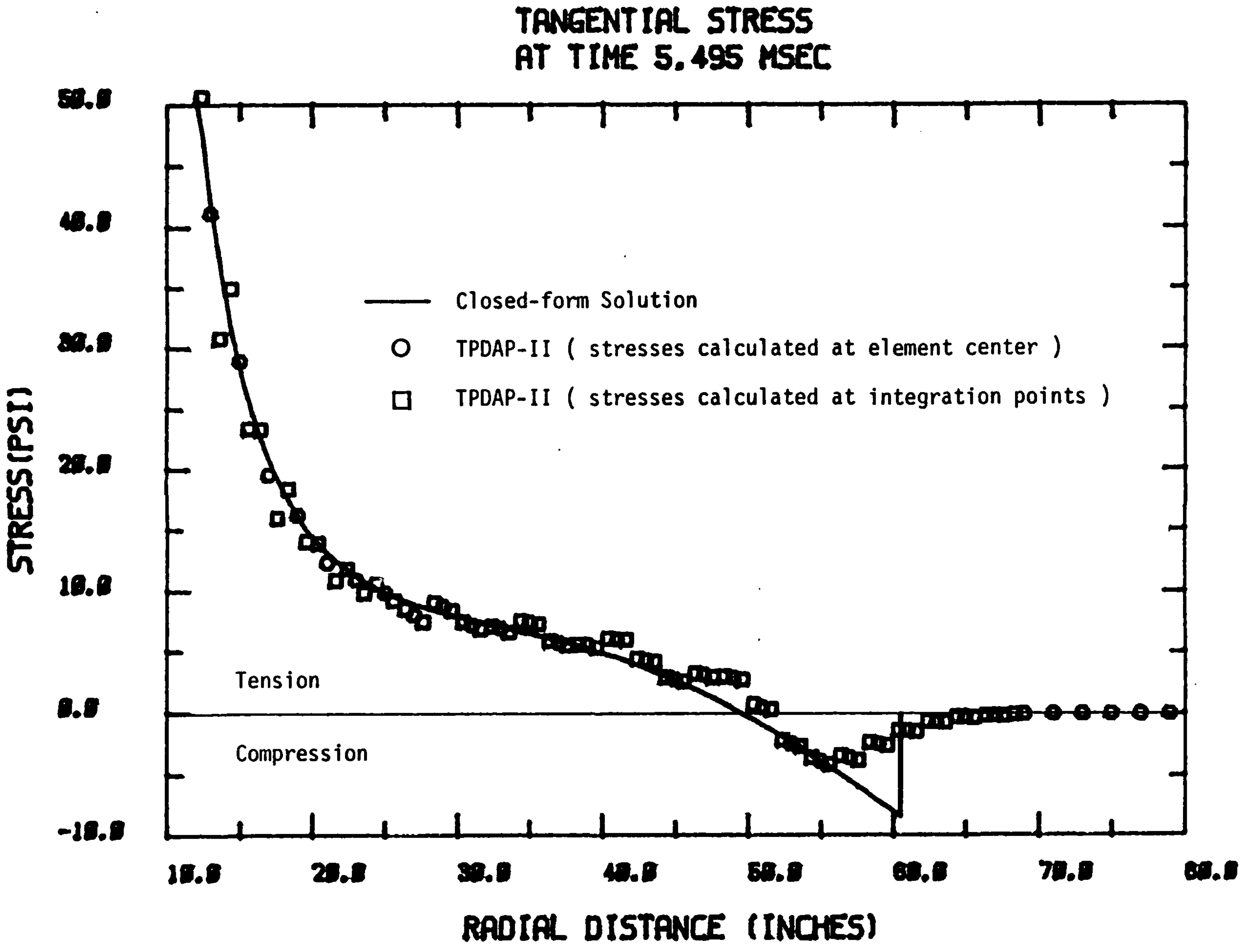
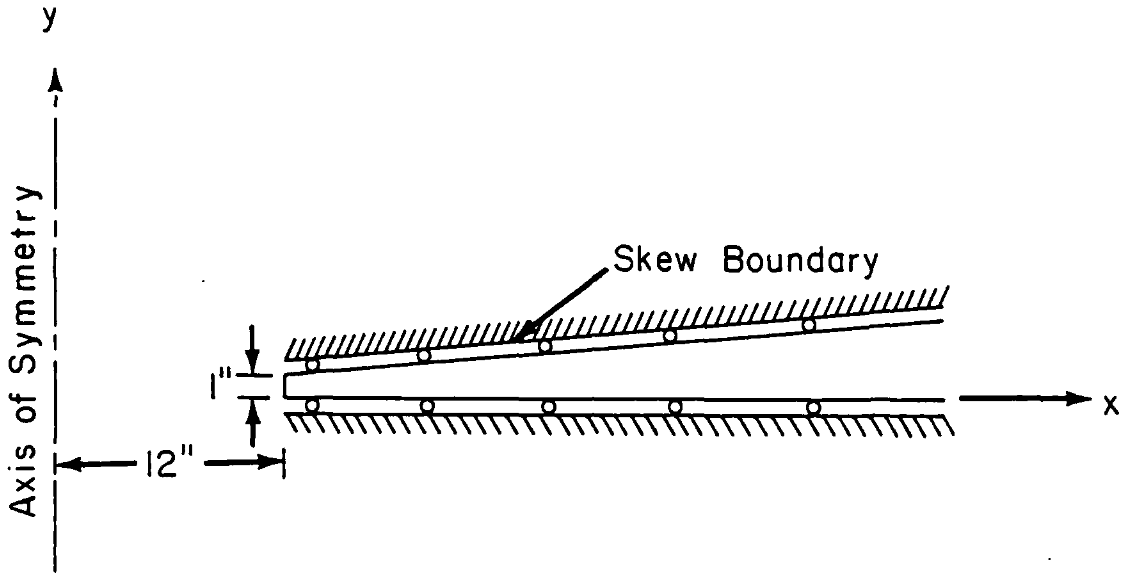


Figure 6.9. Tangential stress profile, verification problem 1.



Material properties, loading conditions and cavity geometry are the same as those of verification problem 1. Stresses and strains are calculated at the center of the element.

Figure 6.10. Verification problem 2, use of skew boundary to simulate elastic spherical wave propagation.

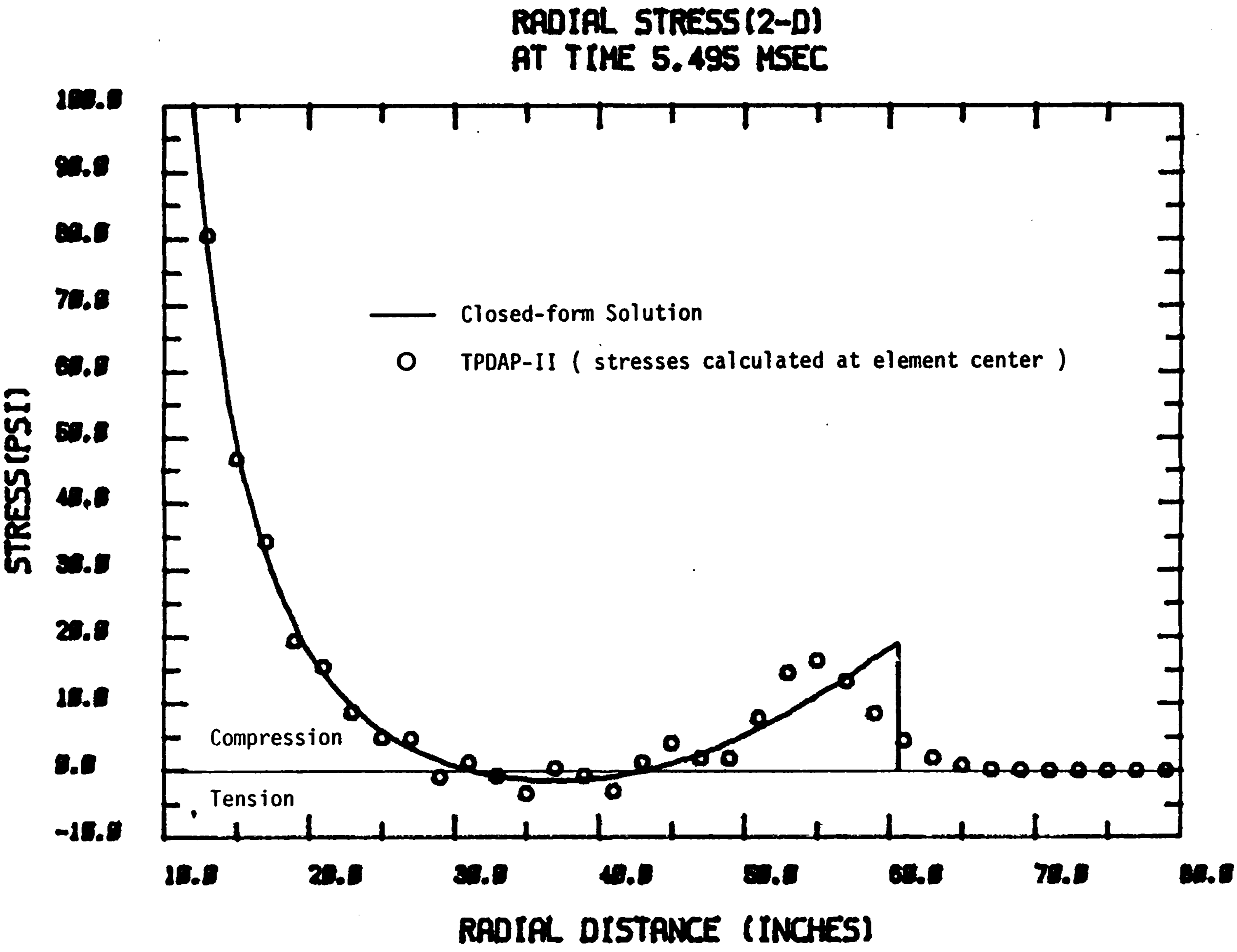


Figure 6.11. Radial stress profile, verification Problem 2.

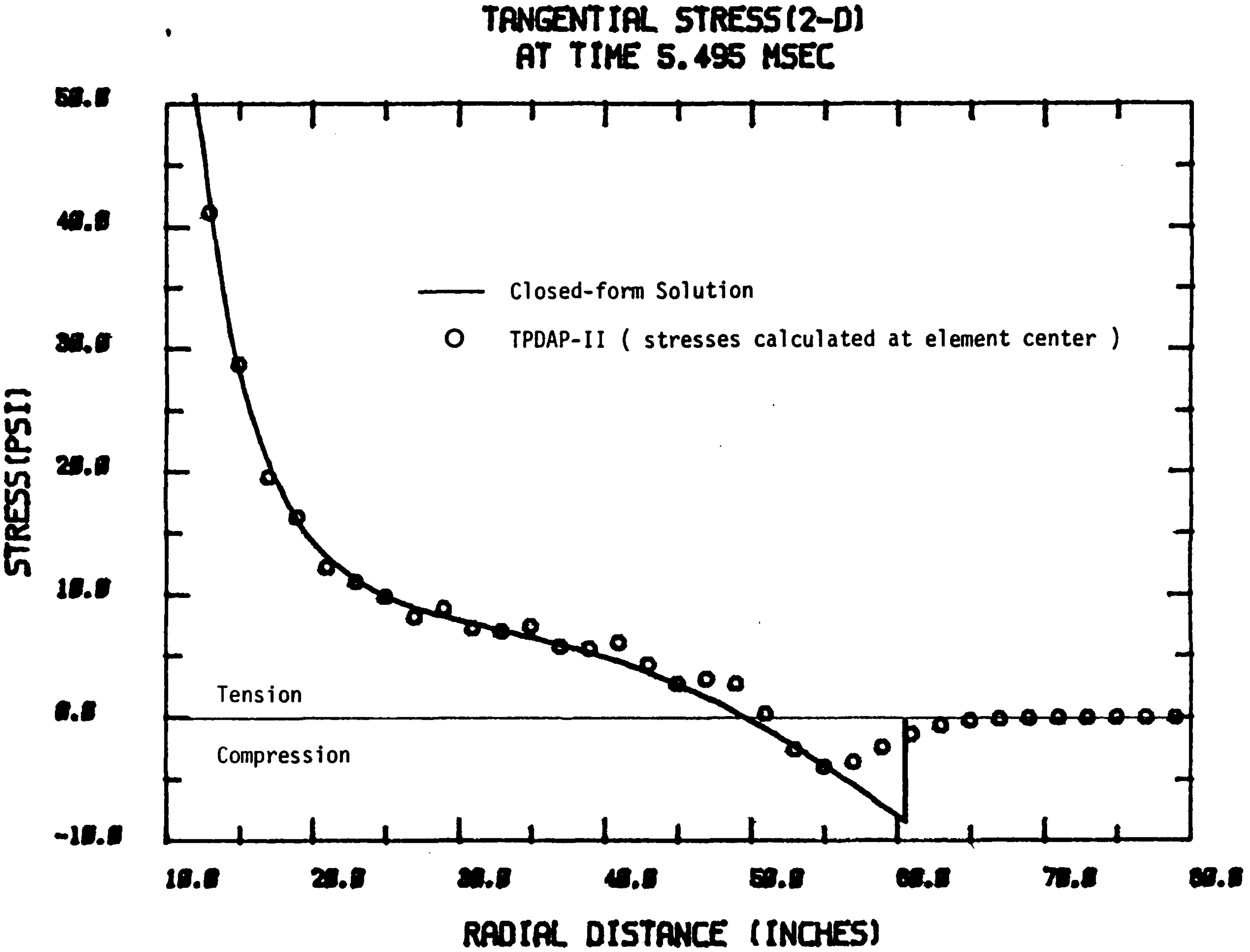
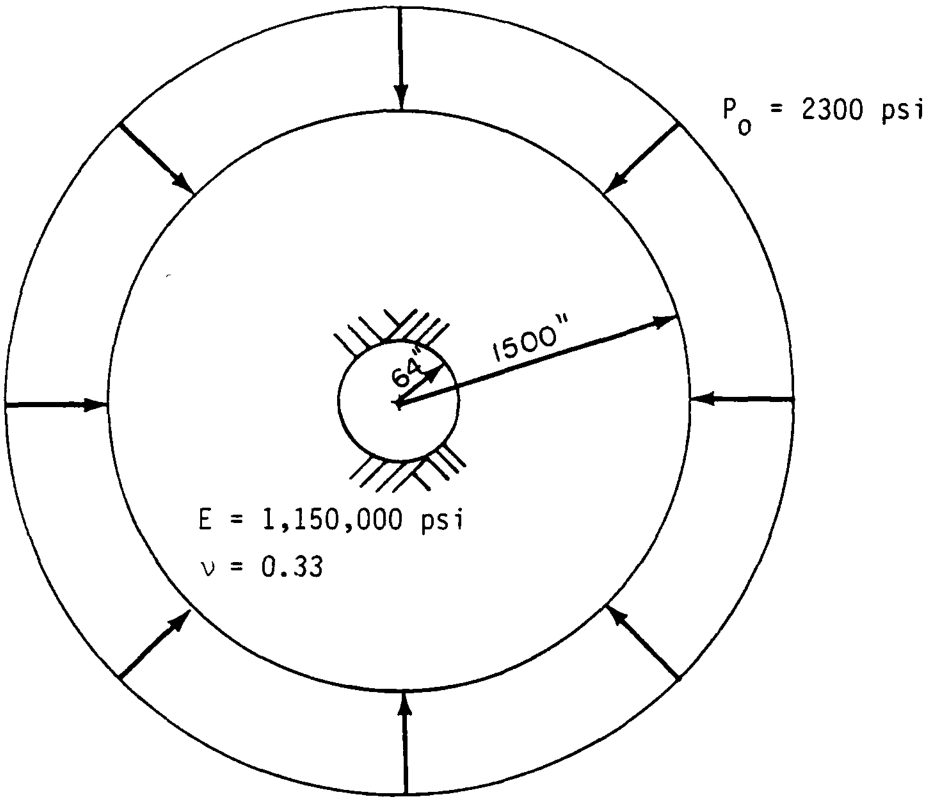


Figure 6.12. Tangential stress profile, verification problem 2.



Material Model: ARA2D

$n = 1$

$\alpha = 1000$ psi

$\beta = 0.689$

$k = 386.6$ psi

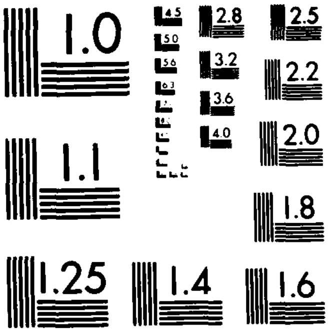
$K = 1$

$\sigma_c = 1800$ psi

$\phi = 18°$

Figure 6.13. Verification problem 3, static nonlinear tunnel analysis in one-phase medium.

AD-A174 749 EXPERIMENTAL AND THEORETICAL RESPONSE OF MULTIPHASE 2/4
POROUS MEDIA TO DYNAM (U) APPLIED RESEARCH ASSOCIATES
INC SOUTH ROYALTON VT NEW ENGLAND K J KIM ET AL
UNCLASSIFIED 13 AUG 86 ARA-5967-86 AFOSR-TR-86-0665 F/G 8/13 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

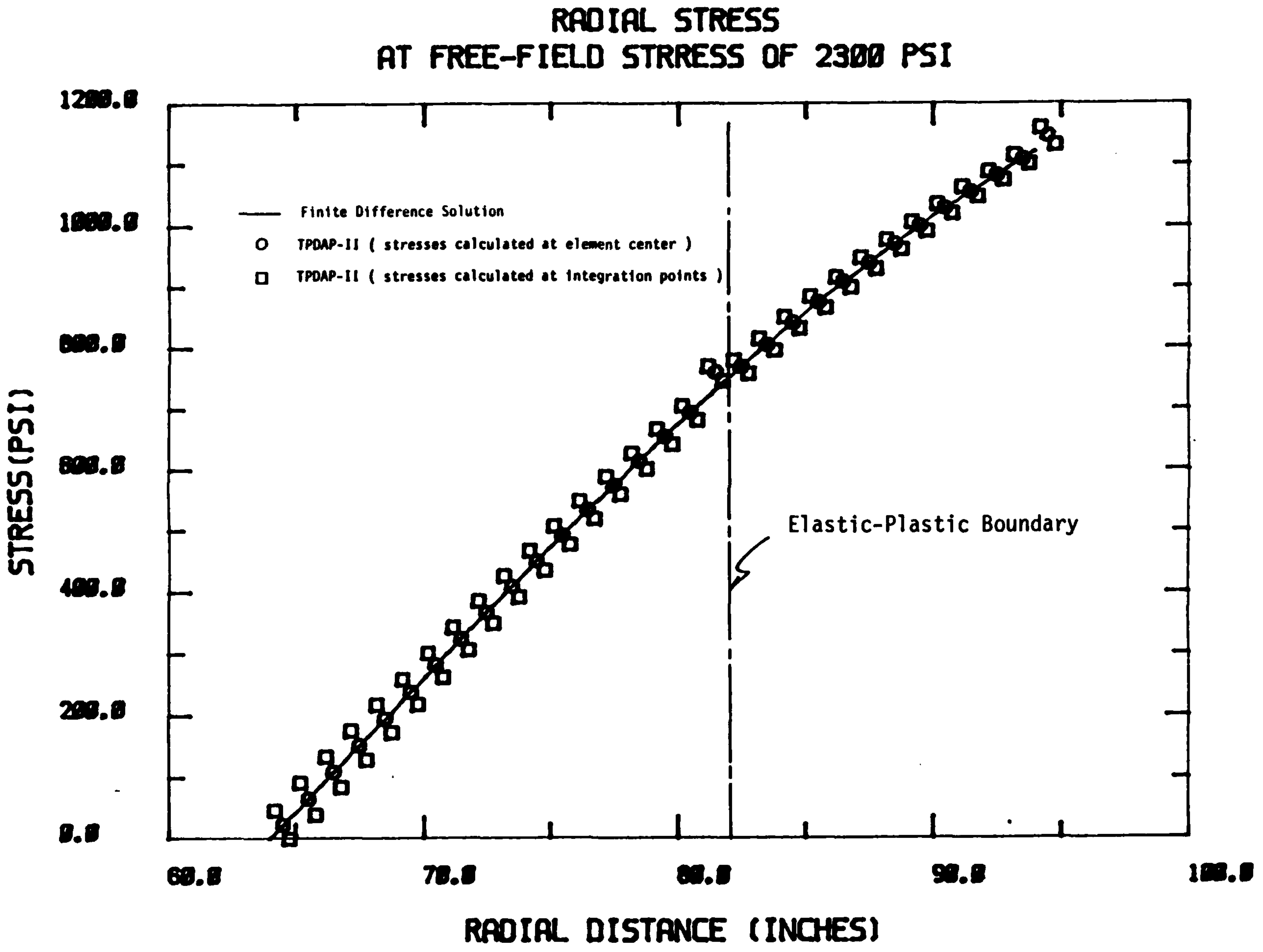


Figure 6.14. Radial stress profile, verification problem 3.

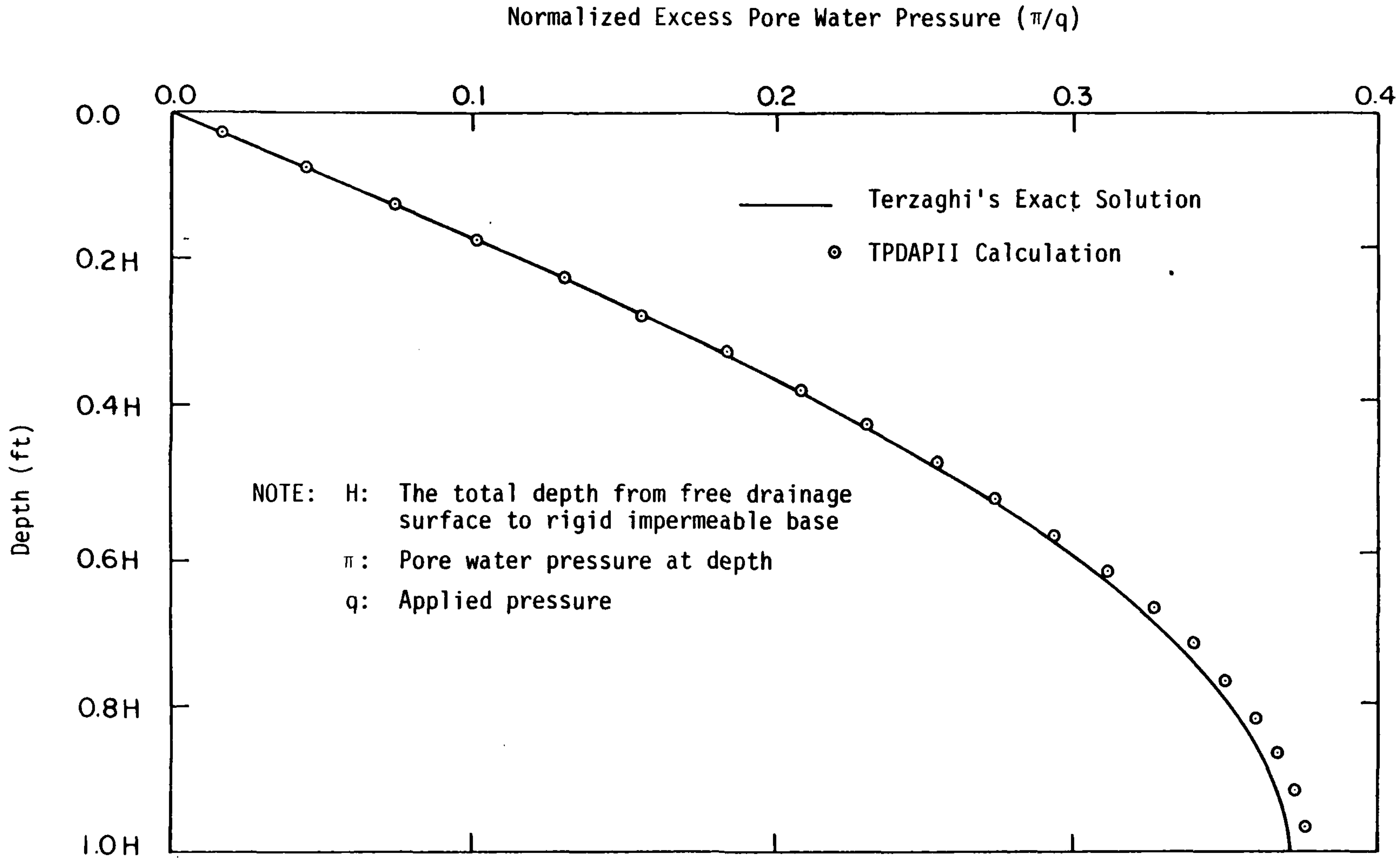
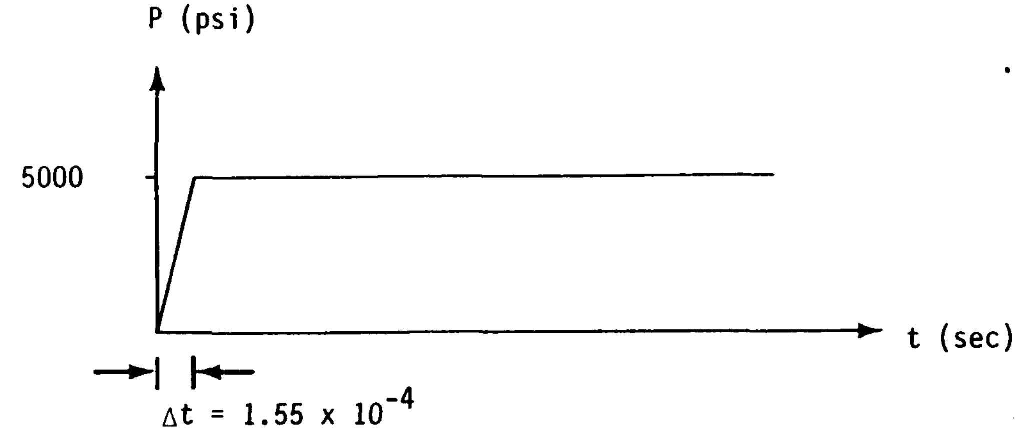


Figure 6.15. Normalized excess pore water pressure profiles at time factor T = 0.5, verification problem 4.



ASSUMED MATERIAL PROPERTIES

Pore Water	
Bulk Modulus	0.29×10^6 psi
Solid Grains	
Bulk Modulus	5.0×10^6 psi
Specific Gravity	2.67
Solid Skeleton	
Bulk Modulus	3000 psi
Constrained Modulus	6000 psi
Poisson's Ratio	0.2
Porosity	0.35
Permeability	0.1 in/sec

Figure 6.16. Verification problem 5, loading time history and material properties used in 1-D plane strain elastic wave propagation through two-phase medium.

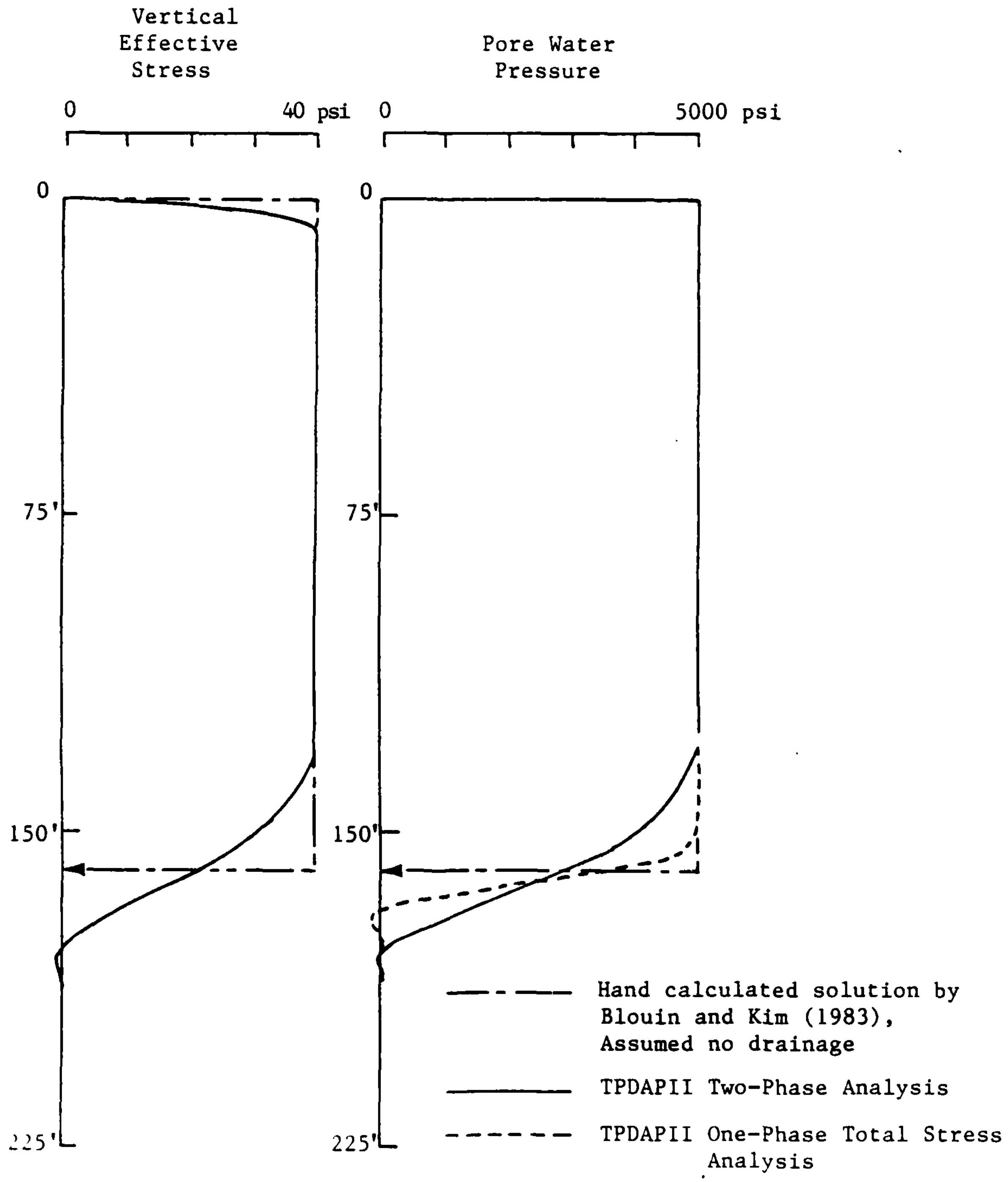


Figure 6.17. Verification problem 5, profiles of the vertical effective stress and pore water pressures at 31 msec.

SECTION 7

THEORETICAL FORMULATIONS FOR THE THREE-PHASE MODEL AND COMPUTATIONAL ALGORITHM

7.1 INTRODUCTION

The theoretical formulations to be used in the three-phase code MPDAP have been completed and are presented in subsection 7.2. The principal specific features of these formulations include:

- a generalized interim fluid friction equation incorporating Biot's theoretical and Ward's empirical results. This is described in Appendix E;
- use of pore pressure as a substitute for relative fluid displacement. This eliminates one degree of freedom in two-dimensional calculations and two degrees of freedom in three-dimensional calculations, resulting in large reductions in running time and storage requirements;
- fully coupled material models;
- partial saturation (three-phase formulation);
- constitutive relations for nonlinear materials

7.2 FORMULATIONS

7.2.1 Notation

Note that positive signs have been used for elongation and tension. A comma denotes differentiation with respect to the subsequent indices and the superposed dot denotes time rate.

$\{u\}$	:	solid phase displacement
$\{U\}$	:	fluid phase displacement
$\{w\}$	:	apparent fluid displacement relative to solid skeleton
$\{\sigma\}$	:	total stress
$\{\sigma'\}$	:	effective stress

π	:	fluid pressure
$\{\epsilon\}$	:	solid phase strain
ϵ_V	:	solid phase volumetric strain
ϵ_F	:	fluid phase volumetric strain
$\{\bar{u}\}_e$	:	solid phase element nodal displacement vector
$\{\bar{\pi}\}_e$	:	element nodal fluid pressure
$\{\bar{u}\}$	:	solid phase structural nodal displacement vector
$\{\bar{\pi}\}$	:	structural nodal fluid pressure
$\{T\}$	:	applied boundary traction
$\hat{Q}$	:	specified boundary flux
$\{b\}$	:	body force vector
k	:	coefficient of permeability
$[D^{ep}]$	:	elasto plastic stress-strain matrix
$\{1\}$	:	unit vector $\{1\}^T = \langle 1\ 1\ 1\ 0\ 0\ 0 \rangle$
n	:	porosity
C_W	:	fluid compressibility
C_{aw}	:	compressibility of air-water mixture
C_G	:	compressibility of solid grains
α	:	compressibility of soil-water mixture
ρ	:	bulk mass density of mixture
ρ_f	:	fluid mass density
γ_f	:	unit weight of the pore fluid
r	:	mass increment factor
β_f	:	Ward's constant which depends on fluid properties
β, γ	:	parameters in Newmark's β method
θ	:	parameter in Wilson's θ method
δ_{ij}	:	Kronecker's delta
$[K_T]$	:	tangent stiffness matrix
$[C]$	:	coupling matrix between solid skeleton and pore fluid
$[E]$	:	matrix of compressibility of pore fluid
$[H]$	:	dissipation resistance matrix
$\{F\}$	:	vector of nodal forces
$\{R\}$	:	internal resisting force vector
$\{\bar{Q}\}$	:	equivalent flow vector

7.2.2 Field Equations

Principle of effective stress:

$$\sigma_{ij} = \sigma'_{ij} + \delta_{ij}\,\pi \qquad (7\text{-}1)$$

Constitutive law for solid skeleton:

$$\{d\sigma'\} = [D^{ep}]\left(\{d\epsilon\} - \frac{C_G}{3}\{1\}d\pi\right) \qquad (7\text{-}2)$$

Continuity equation:

Based on the conservation of mass, the coupled continuity equation of flow, as derived by Kim (1982), is given by

$$(1-n)\, d\epsilon_V + n\, d\epsilon_F - \alpha\, d\pi - (1-n)\, C_G'\, dp' = 0 \qquad (7\text{-}3)$$

where

$$\alpha = n\, C_W + (1-n)\, C_G \qquad (7\text{-}4)$$

and

$$C_G' = \frac{C_G}{(1-n)} \qquad (7\text{-}5)$$

Substitution of Equation 7-5 into Equation 7-3 gives

$$n(d\epsilon_F - d\epsilon_V) = \alpha\, d\pi + C_G\, dp' - d\epsilon_V \qquad (7\text{-}6)$$

Using equation 7-2, the right hand side of Equation 7-6 can be expressed as

$$\boxed{n(d\epsilon_F - d\epsilon_V) = \left(\alpha - \frac{C_G^{\,2}}{9}\{1\}^T[D^{ep}]\{1\}\right) d\pi - \left(\{1\}^T - \frac{C_G}{3}\{1\}^T[D^{ep}]\right)\{d\epsilon\}} \qquad (7\text{-}7)$$

Equation of motion for the bulk mixture:

The governing differential equation for the bulk mixture is expressed as

$$\boxed{\sigma_{ij,j} = \rho\ \ddot{u}_i + \rho_f\ \ddot{w}_i} \qquad (7\text{-}8)$$

Equation of motion for the pore fluid:

The most generalized form of the fluid flow equation is presented in Appendix E and is given by

$$\boxed{\pi_{,i} = \rho_f\ \ddot{U}_i + \frac{1}{k'}\ \dot{w}_i + \frac{r}{n}\ \rho_f\ \ddot{w}_i} \qquad (7\text{-}9)$$

where

$$k' = \frac{k}{\gamma_f \left(1 + \dfrac{\beta_f}{\gamma_f}\ k^{1/2}\ |\dot{w}_i|\right)} \qquad (7\text{-}10)$$

7.2.3 Spatial Discretization

Within each element, field variables can be discretized into element nodal values.

$$\begin{aligned} \{u\} &= [N]\ \{\bar{u}\}_e \\ \{\epsilon\} &= [B]\ \{\bar{u}\}_e \\ \pi &= \langle G\rangle\ \{\bar{\pi}\}_e \\ \{\pi_{,i}\} &= [A]\ \{\bar{\pi}\}_e \end{aligned} \qquad (7\text{-}11)$$

7.2.4 Incremental Relationships of Field Variables

Stress vector at time step n can be expressed as

$$\{\sigma_n\} = \{\sigma_{n-1}\} + \{\Delta\sigma'_n\} + \{1\}\ \Delta\pi_n \qquad (7\text{-}12)$$

Substitution of Equation 7-2 into Equation 7-12 yields

$$\boxed{\{\sigma_n\} = \{\sigma_{n-1}\} + [D^{ep}]\{\Delta\epsilon_n\} + \left(\{1\} - \frac{C_G}{3}\,[D^{ep}]\{1\}\right)\Delta\pi_n} \qquad (7\text{-}13)$$

7.2.5 Assumptions On Fluid Flow Equations

To simplify mathematical derivations, the following assumptions are made relative to the motion of fluid between the time step n-1 and n:

$$k' = k'_{n-1} = \text{Constant} \qquad (7\text{-}14)$$

and

$$\rho_f\ddot{U}_i + \frac{r}{n}\,\rho_f\ddot{w}_i = \left(\rho_f\ddot{U}_i + \frac{\dot{r}}{n}\;\rho_f\ddot{w}_i\right)_{n-\frac{1}{2}} = \text{Constant} \qquad (7\text{-}15)$$

Now, substituting Equations 7-14 and 7-15 into Equation 7-9 and differentiating with respect to time,

$$\dot{\pi}_{,i} = \frac{1}{k'_{n-1}}\,\ddot{w}_i \qquad (7\text{-}16)$$

or

$$\ddot{w}_i = k'_{n-1}\,\dot{\pi}_{,i} \qquad (7\text{-}17)$$

and

$$\boxed{\ddot{U}_i = \ddot{u}_i + \frac{1}{n}\,k'_{n-1}\,\dot{\pi}_{,i}} \qquad (7\text{-}18)$$

Rewriting Equation 7-9 as assumed,

$$\pi_{,i} = \left(\rho_f\ddot{U}_i + \frac{r}{n}\,\rho_f\ddot{w}_i\right)_{n-\frac{1}{2}} + \frac{1}{k'_{n-1}}\,\dot{w}_i \qquad (7\text{-}19)$$

Now, backsubstituting Equations 7-17 and 7-18 into Equation 7-19 and rearranging,

$$\boxed{\begin{gathered} \dot{w}_i = k'_{n-1}\,\pi_{,i} - \left(\rho_f\, k'_{n-1}\,\ddot{u}_i + \frac{\rho_f}{n}\,(1+r)(k'_{n-1})^2\,\dot{\pi}_{,i}\right)_{n-\frac{1}{2}} \\ t_{n-1} \leqslant t \leqslant t_n \end{gathered}} \tag{7-20}$$

Thus Equations 7-18 and 7-20 express the absolute and relative fluid motions respectively in terms of skeleton motion and the gradient of pore fluid pressure.

7.2.6 Global Equilibrium Equation For The Bulk Medium

Shown schematically in Figure 7.1 is the total stresses and virtual displacements on the boundary of the infinitesimal element. The total stresses are in equilibrium with the applied boundary tractions. Taking the solid skeleton movement as the virtual displacement, the internal and external virtual work must be equal. The internal virtual work at time step n ($t = t_n$) is given by

$$\delta W_I = \int_V \{\delta\epsilon\}^T\{\sigma_n\}\;dv \tag{7-21}$$

Substituting Equations 7-11 and 7-13 into Equation 7-21,

$$\begin{aligned} \delta W_I = \{\delta\bar{u}\}^T \Bigg[& \left(\sum \int_V [B]^T[D^{ep}][B]dv\right)\{\Delta\bar{u}\}_n \\ & + \left(\sum \int_V [B]^T\left(\{1\} - \frac{C_G}{3}\,[D^{ep}]\{1\}\right)\,\langle G\rangle dv\right)\{\Delta\bar{\pi}\}_n \\ & + \sum \int_V [B]^T\{\sigma_{n-1}\}\;dv \Bigg] \end{aligned} \tag{7-22}$$

The external virtual work at time step n is given by

$$\delta W_E = -\int_V \{\delta u\}^T\{F_I\}_n + \int_s \{\delta u\}^T\{T\}_n \, ds \qquad (7\text{-}23)$$

$$= \{\delta\bar{u}\}^T\left[-\left(\sum \int_V [N]^T(\rho - n\rho_f)[N] \, dv\right)\{\ddot{\bar{u}}\}_n\right.$$

$$\left. - \left(\sum \int_V [N]^T \, n\rho_f\{\ddot{U}\}_n \, dv\right) + \sum \int_s [N]^T\{T\}_n \, ds \right]$$

Substituting Equation 7-18 into Equation 7-23 yields

$$\delta W_E = \{\delta\bar{u}\}^T \left[-\left(\sum \int_V [N]^T \rho[N] \, dv\right) \{\ddot{\bar{u}}\}_n\right. \qquad (7\text{-}24)$$

$$- \left(\sum \int_V [N]^T\rho_f k'[A] \, dv\right) \{\ddot{\bar{\pi}}\}_n$$

$$\left. + \left(\sum \int_s [N]^T\{\dot{T}\}_n \, ds\right)\right]$$

Since internal and external virtual work are equal,

$$\delta W_I = \delta W_E \qquad (7\text{-}25)$$

Now, global equilibrium equations for the bulk mixture are obtained by substituting Equations 7-22 and 7-24 into Equation 7-25.

$$[M_t]\{\ddot{\bar{u}}\}_n + [M_c]\{\dot{\bar{\pi}}\}_n + [K_T]\{\Delta\bar{u}\}_n + [C]\{\Delta\bar{\pi}\}_n \qquad (7\text{-}26)$$

$$= \{F\}_n - \sum \int_V [B]^T\{\sigma_{n-1}\} \, dv$$

where

$$[M_t] = \sum \int_V [N]^T \rho [N] \; dv$$

$$[M_c] = \sum \int_V [N]^T \rho_f k' [A] \; dv$$

$$[K_T] = \sum \int_V [B]^T [D^{ep}] [B] \; dv \qquad (7\text{-}27)$$

$$[C] = \sum \int_V [B]^T \left(\{1\} - \frac{C_G}{3} [D^{ep}] \{1\} \right) <G> \; dv$$

$$\{F\}_n = \sum \int_S [N]^T \{T\}_n \; ds$$

Substitution of Equation F-6 into Equation 7-26 yields

$$\boxed{[M_t]\{\ddot{\bar{u}}\}_n + \frac{\Delta t}{2} [M_c]\{\ddot{\bar{\pi}}\}_n + [K_t]\{\Delta\bar{u}\}_n + [C]\{\Delta\bar{\pi}\}_n = \{P_u\}_n} \qquad (7\text{-}28)$$

where

$$\{P_u\}_n = \{F\}_n - \sum \int_V [B]^T \{\sigma_{n-1}\} \; dv - [M_c] \left(\{\dot{\bar{\pi}}\}_{n-1} + \frac{\Delta t}{2} \{\ddot{\bar{\pi}}\}_{n-1} \right) \qquad (7\text{-}29)$$

7.2.7 Global Equilibrium Equations for Pore Fluid

Figure 7.2 shows schematically the complementary virtual pore pressures, skeleton and fluid velocities on the boundary of the infinitesimal element. The internal fluid movements relative to the solid skeleton are compatible with the specified boundary flux. Thus, taking the complementary virtual stresses as the pore pressure field at a certain time, t, the internal and external complementary virtual work which is done between time t_{n-1} and t_n must be the same. That is,

$$\delta W_I{}^* = \delta W_E{}^* \tag{7-30}$$

where the internal complementary work is given by

$$\delta W_I{}^* = \int_{t_{n-1}}^{t_n} \int_V \delta\pi \; n \frac{\partial}{\partial t}(\epsilon_F - \epsilon_V) \; dv \; dt \quad t \int_{t_{n-1}}^{t_n} \int_V \{\delta\pi,_i\}^T\{\dot{w}\} \; dv \; dt \tag{7-31}$$

and the external complementary work by

$$\delta W_E{}^* = \int_{t_{n-1}}^{t_n} \int_s \delta\pi \; \hat{Q} \; ds \; dt \tag{7-32}$$

Substitution of Equations 7-7 and 7-20 into Equation 7-31 yields

$$\begin{aligned}\delta W_I{}^* = \int_{t_{n-1}}^{t_n} \Bigg[& \int_V \delta\pi^T \left(\alpha - \frac{C_G{}^2}{9}\{1\}^T[D^{ep}]\{1\}\right)\dot{\pi} \; dv \\ & - \int_V \delta\pi^T \left(\{1\}^T - \frac{C_G}{3}\{1\}^T[D^{ep}]\{\dot{\epsilon}\}\right) dv \\ & + \int_V \{\delta\pi,_i\}^T \; k'\{\pi,_i\} \; dv \\ & - \int_V \{\delta\pi,_i\}^T \left(\rho_f k'\{\ddot{u}_i\} + \frac{\rho_f}{n}(1+r)(k')^2\{\dot{\pi},_i\}_{n-\frac{1}{2}}\right) dv \Bigg] \; dt\end{aligned} \tag{7-33}$$

Note that the last integral term in Equation 7-33 is assumed to be constant with respect to time during the step. Discretizing Equation 7-33 by Equation 7-11 yields

$$\delta W_I{}^{*} = \{\delta\bar{\pi}\}^T \Bigg[[E]\{\Delta\bar{\pi}\}_n - [C]^T\{\Delta\bar{u}\}_n + \int_{t_{n-1}}^{t_n} [H]\{\bar{\pi}\}\, dt \qquad (7\text{-}34)$$

$$- \Delta t\ [M_c]^T\{\ddot{\bar{u}}\}_{n-½} - \Delta t[M_f]\{\dot{\bar{\pi}}\}_{n-½} \Bigg]$$

where

$$[E] = \sum \int_V <G>^T \left(\alpha - \frac{C_G{}^2}{9}\ \{1\}^T[D^{ep}]\{1\} \right) <G>\ dv \qquad (7\text{-}35)$$

$$[H] = \sum \int_V [A]^T\ k'[A]\ dv$$

$$[M_f] = \sum \int_V [A]^T \rho_f\ \frac{(1+r)}{n}\ (k')^2\ [A]\ dv$$

Let

$$\{\ddot{\bar{u}}\}_{n-½} = ½\left(\{\ddot{\bar{u}}\}_n + \{\ddot{\bar{u}}\}_{n-1}\right) \qquad (7\text{-}36)$$

Then, substituting Equations F-5, F-9 and 7-36 into Equation 7-34,

$$\delta W_I{}^{*} = \{\delta\bar{\pi}\}^T \Bigg[[E]\{\Delta\bar{\pi}\}_n - [C]^T\{\Delta\bar{u}\}_n \qquad (7\text{-}37)$$

$$+ [H] \left(\Delta t\ \{\bar{\pi}\}_{n-1} + \frac{\Delta t^2}{2}\{\dot{\bar{\pi}}\}_{n-1} + \frac{\Delta t^3}{12}\{\ddot{\bar{\pi}}\}_{n-1} + \frac{\Delta t^3}{12}\{\ddot{\bar{\pi}}\}_n \right)$$

$$- \frac{\Delta t}{2}[M_c]^T \left(\{\ddot{\bar{u}}\}_n + \{\ddot{\bar{u}}\}_{n-1} \right)$$

$$- \Delta t\ [M_f] \left(\{\dot{\bar{\pi}}\}_{n-1} + \frac{\Delta t}{4}\{\ddot{\bar{\pi}}\}_{n-1} + \frac{\Delta t}{4}\{\ddot{\bar{\pi}}\}_n \right) \Bigg]$$

For the external complementary virtual work, discretizing Equation 7-32 by Equation 7-11 yields

$$\delta W_E^{\star} = \{\delta\bar{\pi}\}^T \left[\int_{t_{n-1}}^{t_n} \{Q\}\ dt \right] \tag{7-38}$$

where

$$\{Q\} = \sum \int_s \langle G\rangle^T\ \hat{Q}\ ds \tag{7-39}$$

Assuming a linear variation of $\{Q\}$ between time t_{n-1} and t_n, Equation 7-38 can be expressed as

$$\delta W_E^{\star} = \{\delta\bar{\pi}\}^T \left[\frac{\Delta t}{2} \left(\{Q\}_{n-1} + \{Q\}_n\right)\right] \tag{7-40}$$

Now, substituting Equations 7-37 and 7-40 into Equation 7-30, the following global equilibrium equations for the pore fluid are obtained:

$$\boxed{\begin{aligned}&\frac{\Delta t}{2}[M_c]^T\{\ddot{\bar{u}}\}_n + \left(\frac{\Delta t^2}{4}[M_f] - \frac{\Delta t^3}{12}[H]\right)\{\ddot{\bar{\pi}}\}_n \\ &+ [C]^T\{\Delta\bar{u}\}_n - [E]\{\Delta\bar{\pi}\}_n = \{P_\pi\}_n\end{aligned}} \tag{7-41}$$

where

$$\{P_\pi\}_n = [H]\left(\Delta t\{\bar{\pi}\}_{n-1} + \frac{\Delta t^2}{2}\{\dot{\bar{\pi}}\}_{n-1} + \frac{\Delta t^3}{12}\{\ddot{\bar{\pi}}\}_{n-1}\right) \tag{7-42}$$

$$- \frac{\Delta t}{2}\,[M_c]^T\{\ddot{\bar{u}}\}_{n-1} - \Delta t[M_f]\left(\{\dot{\bar{\pi}}\}_{n-1} + \frac{\Delta t}{4}\{\ddot{\bar{\pi}}\}_{n-1}\right)$$

$$- \frac{\Delta t}{2}\left(\{Q\}_{n-1} + \{Q\}_n\right)$$

7.2.8 Combined Global Equilibrium Equations

Equations 7-28 and 7-41 can be combined in the following matrix form:

$$\boxed{[M]\{\ddot{\bar{d}}\}_n + [K]\{\Delta\bar{d}\}_n = \{\bar{P}\}_n} \tag{7-43}$$

where

$$[M] = \left[\begin{array}{c|c} [M_t] & \frac{\Delta t}{2}\,[M_c] \\ \hline \frac{\Delta t}{2}[M_c]^T & \frac{\Delta t^2}{4}[M_f] - \frac{\Delta t^3}{12}[H] \end{array}\right] \tag{7-44}$$

$$[K] = \left[\begin{array}{c|c} [K_T] & [C] \\ \hline [C]^T & -[E] \end{array}\right]$$

$$\{P\}_n = \left\{ \begin{array}{c} \{P_u\}_n \\ -\,-\,-\,- \\ \{P_\pi\}_n \end{array} \right\}$$

$$\{\ddot{\bar{d}}\}_n = \left\{ \begin{array}{c} \{\ddot{\bar{u}}\}_n \\ -\,-\,-\,- \\ \{\ddot{\bar{\pi}}\}_n \end{array} \right\}$$

$$\{\Delta\bar{d}\}_n = \left\{ \begin{array}{c} \{\Delta\bar{u}\}_n \\ -\,-\,-\,- \\ \{\Delta\bar{\pi}\}_n \end{array} \right\}$$

7.2.9 Linearized Global Equilibrium Equations

Introducing a time integration method which incorporates both Newmark's β method and Wilson's θ method, the generalized acceleration vector is expressed as

$$\{\ddot{\bar{d}}\}_n = C_1\{\Delta\bar{d}\}_n + C_2\{\dot{\bar{d}}\}_{n-1} + C_3\{\ddot{\bar{d}}\}_{n-1} \tag{7-45}$$

where

$$C_1 = \frac{1}{\beta\theta^3\Delta t^2}$$

$$C_2 = -\frac{1}{\beta\theta^2\Delta t} \tag{7-46}$$

$$C_3 = 1 - \frac{1}{2\beta\theta}$$

and the generalized velocity vector is expressed as

$$\{\dot{\bar{d}}\}_n = B_1\ \{\Delta\bar{d}\}_n + B_2\ \{\dot{\bar{d}}\}_{n-1} + B_3\ \{\ddot{\bar{d}}\}_{n-1} \tag{7-47}$$

where

$$B_1 = \frac{\gamma}{\beta\theta^3\Delta t} \qquad (7\text{-}48)$$

$$B_2 = 1 - \frac{\gamma}{\beta\theta^2}$$

$$B_3 = 1 - \frac{\gamma}{2\beta\theta}\ \Delta t$$

Substituting Equation 7-45 into Equation 7-43 and rearranging, we can obtain the following linearized global equilibrium equations which can be solved simultaneously at each step:

$$\boxed{[\tilde{K}]\{\Delta\bar{d}\}_n = \{\tilde{P}\}_n} \qquad (7\text{-}49)$$

where

$$[\tilde{K}] = C_1[M] + [K] \qquad (7\text{-}50)$$

$$\{\tilde{P}\}_n = \{\bar{P}\}_n - [M]\left(C_2\{\dot{\bar{d}}\}_{n-1} + C_3\{\ddot{\bar{d}}\}_{n-1}\right)$$

7.3 PARTIAL SATURATION

In numerous geologic settings the soil or rock is not fully saturated. Rischbieter et al. (1977) demonstrated that even a minute amount of entrapped air drastically alters the pore pressure response in multiphase porous materials. Thus, a complete treatment of multiphase media should include the capability of calculating stress wave propagation and pore fluid response in three-phase porous materials.

Kim (1982) developed a unique formulation for the compressibility of the air-water mixture in partially saturated porous media. This formulation has been extensively applied in quasi-static problems and verified against experimental data. It will be incorporated into MPDAP to provide it with three-phase dynamic capability.

The compressibility, C_{aw}, of the air-water mixture in partially saturated media is given by

$$C_{aw} = (1 - S_o + H_c S_o) \frac{\pi_{ao}}{(\pi + T)^2} \qquad (7\text{-}51)$$

where

S_o = initial degree of saturation

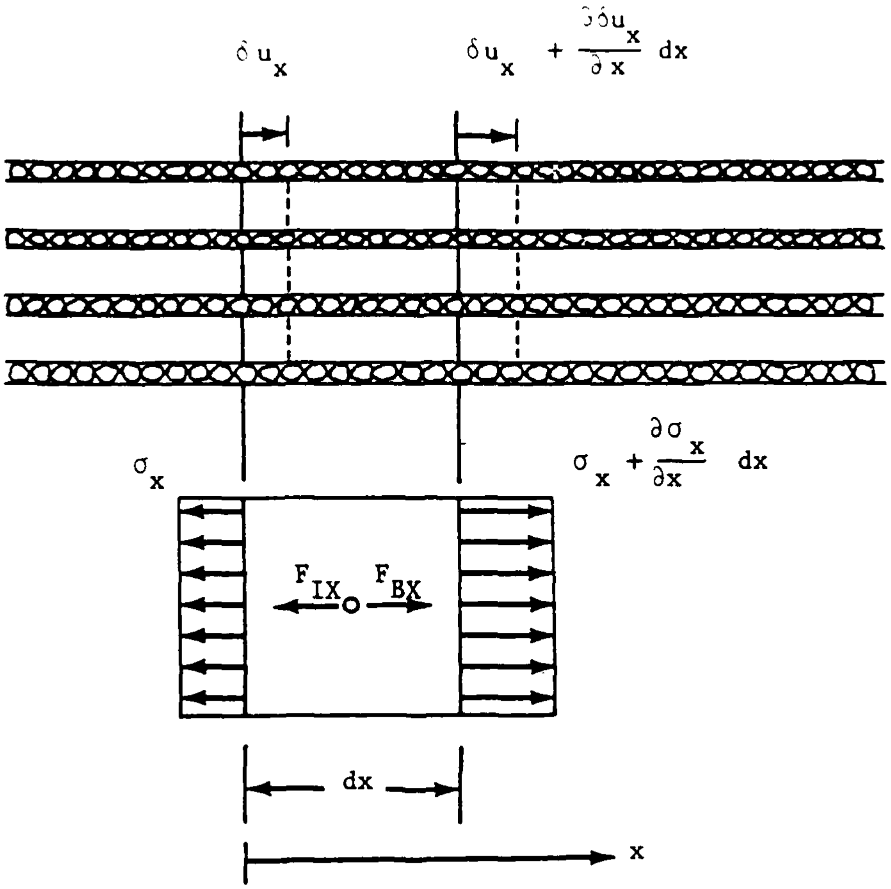
H_c = coefficient of solubility (Henry's constant)

π_{ao} = initial pore air pressure (absolute)

π = current pore water pressure (absolute)

T = pressure difference between the air and pore water due to surface tension

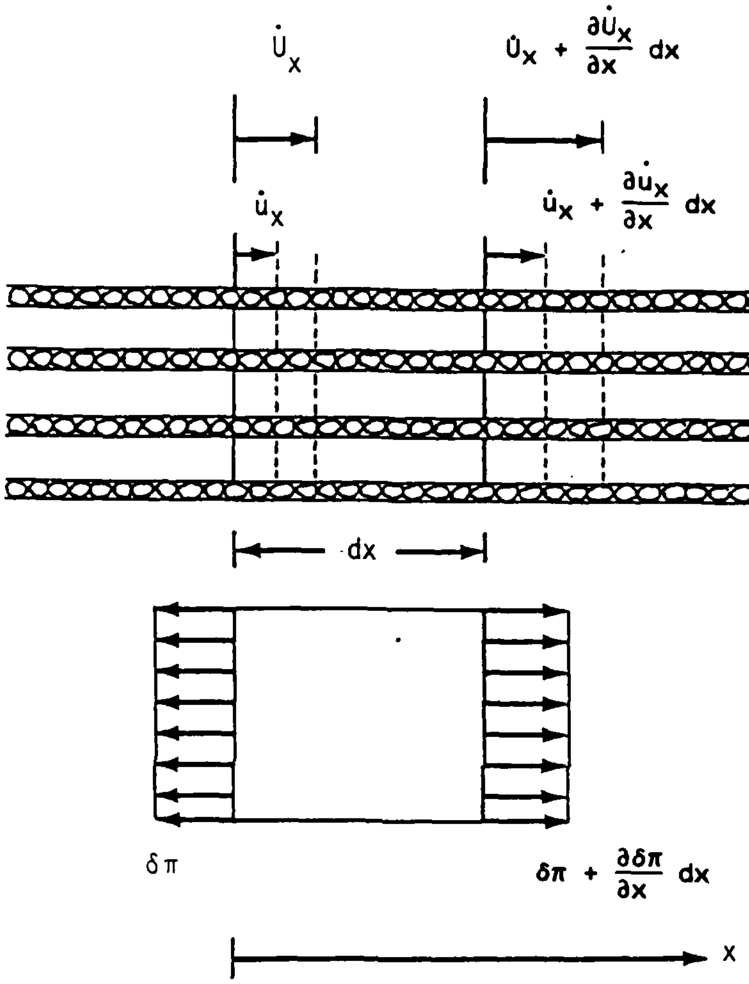
As indicated by Equation 7-51, the compressibility of the air-water mixture is nonlinear with respect to the current pore water pressure.


δu_x
$\delta u_x + \frac{\partial \delta u_x}{\partial x} dx$
σ_x
$\sigma_x + \frac{\partial \sigma_x}{\partial x} dx$
F_{IX}
F_{BX}
dx
x

$$F_{BX} = (\rho\, dV)\, b_x$$

$$F_{IX} = (\rho - n\, \rho_f)\, dV\, \ddot{u}_x + n\, \rho_f\, dV\, \ddot{U}_x$$

Figure 7.1. Total stress and virtual displacement field in x-direction.


$\dot{U}_x$
$\dot{U}_x + \frac{\partial \dot{U}_x}{\partial x} dx$
$\dot{u}_x$
$\dot{u}_x + \frac{\partial \dot{u}_x}{\partial x} dx$
dx
$\delta \pi$
$\delta \pi + \frac{\partial \delta \pi}{\partial x} dx$
x

$$\delta W^{*}_{Ix} = - \delta \pi \; n \; dy \; dz \; (\dot{U}_x - \dot{u}_x) \; dt + (\delta \pi + \frac{\partial \delta \pi}{\partial x} dx) \; n \; dy \; dz$$

$$\left[\; \dot{U}_x + \frac{\partial \dot{U}_x}{\partial x} dx \; - \; \dot{u}_x + \frac{\partial \dot{u}_x}{\partial x} dx \; \right] dt$$

Figure 7.2. Relative fluid velocity and complementary virtual fluid pressure field in x-direction.

SECTION 8

SUMMARY

8.1 LABORATORY WORK

Three major laboratory efforts were undertaken during the first year:

- measurement of grain compressibility and evaluation of the influence of microporosity on grain compressibility;

- development of the shock consolidation test designed to replicate dynamic undrained loadings, liquefaction and subsequent consolidation, and

- design and fabrication of a pressure vessel to measure fluid friction and energy dissipation as functions of flow rate, change in flow rate and pressure gradient in a variety of porous materials over a wide range of pressure gradients.

Section 2 describes the experimental procedure and presents results of the grain compressibility tests. The values of grain bulk modulus measured for steel and quartz were within the range of published values and thus validated the experimental procedure. The modulus value for the Enewetak beach sand was very close to the published value for calcium carbonate, despite the appreciable degree of microporosity in this material. Values for all cored materials from Enewetak were about 30% lower than the beach sand, indicating a possible influence of microporosity. Even so, the overall influence of microporosity will be very slight, since even the reduced solid grain moduli of the cored Enewetak materials are still higher than the moduli of solid quartz grains. Improvements in test resolution in the high and low pressure regimes are being pursued.

The third section describes the development of two types of uniaxial strain shock consolidation tests and presents preliminary test data on cemented and uncemented samples. These tests are designed to simulate response to undrained uniaxial dynamic loadings, the liquefaction which occurs early in the dynamic unloading, and finally the quasi-static consolidation (volume reduction) which occurs as the sample consolidates and the original pretest effective stresses are reestablished. Preliminary test data show that in both the cemented and uncemented test samples liquefaction was achieved shortly after undrained unloading began. Considerable quasi-static consolidation during the late-time drained loadings was measured, typically approximating the peak volume strain achieved during the undrained loading portion of each test. Future testing will examine the influence of both peak stress and different strain paths on the late-time consolidation.

Section 4 describes the design and fabrication of a dynamic flow test device which will be used to develop empirical models for fluid friction. The test device is servo-controlled and is designed to provide measurements of fluid friction and energy dissipation over a wide range of flow conditions. These include flow in the laminar, transition and turbulent flow regimes. The test device can utilize either specified flows or pressure gradients as input to the porous test samples. Initial evaluations of this device are underway using simple circular ducts for comparison to available flow solutions.

8.2 THEORETICAL WORK

Section 5 describes a series of computer algorithms used to predict undrained hydrostatic and uniaxial compressive behavior of fully saturated porous media. Inputs include the drained skeleton effective stress-strain response and the pressure-volume response of the pore water and solid grains. The skeleton response can be obtained directly from drained laboratory hydrostatic and/or uniaxial strain compression tests. The output includes the total undrained stress-strain response as well as the pore pressure and effective stress response.

Four formulations are presented, two based on the mixture model and the others based on the fully coupled model. The formulations based on the mixture model use both engineering strain and true strain. They provide a good approximation of response for materials in which effective stresses are relatively small. The formulations based on the fully coupled material model are expressed in terms of engineering strain and are applicable to all classes of porous materials. The four computer algorithms for the undrained loadings are presented in Appendices A through D along with sample problems.

Section 6 describes the modeling and computational advances incorporated into the new two phase code TPDAPII. Detailed descriptions of the effective stress models are provided. These include the uniaxial strain model (UNIAX), the decoupled elasto-plastic model (DCOUP) and our two-dimensional plasticity model with coupled deviatoric and volumetric behavior (ARA2D). A new technique which uses a single point at each element center to compute strains, stresses and constitutive equations is presented. Use of this technique greatly reduces data storage requirements and computational times, especially for large scale nonlinear finite element analyses. It also eliminates local stress oscillations and numerical instabilities.

The TPDAPII code is presented in Appendix H and a user's manual is included as Appendix G. Five verification problems are included in Section 6 which exercise both the single and two phase options of TPDAPII in static, quasi-static and dynamic problems. Results compare extremely well to known analytical solutions. The single point technique is used in all problems and eliminates oscillations observed in several solutions using conventional calculational techniques.

Section 7 describes the theoretical formulations which will be incorporated into the general multiphase code MPDAP, being written as part of next year's effort. The features of these formulations include:

- a generalized interim fluid friction equation incorporating Biot's theoretical and Ward's empirical results. This is described in Appendix E;

- use of pore pressure as a substitute for relative fluid displacement. This eliminates one degree of freedom in two-dimensional calculations and two degrees of freedom in three-dimensional calculations, resulting in large reductions in running time and storage requirements;

- fully coupled material models;

- partial saturation (three phase formulation);

- constitutive relations for nonlinear materials.

APPENDIX A

UNDRAINED BEHAVIOR OF MIXTURE MODEL BASED ON ENGINEERING STRAIN

The computer program MENGR, listed in subsection A.1 reads a set of pressure-water modulus data from TAPE6. For grain bulk modulus, the empirical equation from Section 5.2.2 is hard-wired in the subroutine BULKG.

DESCRIPTION OF INPUT FILE: TAPE6

Card 1

NWATER (I5)

NWATER: Number of pressure/water modulus pairs

Card 2

$$\left.\begin{array}{l} P_1,\ K_{w1} \\ P_2,\ K_{w2} \\ -\quad - \\ -\quad - \\ P_n,\ K_{wn} \end{array}\right\}$$

NWATER cards with format (2F10.0) for each card. Note that P_i and K_{wi} are in terms of bars.

DESCRIPTION OF INPUT FILE: TAPE7

Card 1

POR, EMAX, NDIV (2F10.0, I5)

POR = Initial Porosity
EMAX = Maximum total strain at which calculation terminates
NDIV = Total number of steps

A.1 List of Program MENGR

```
      PROGRAM MENGR
      COMMON /WATER/ PP(100),BM(100)
C
C     READ PRESSURE-BULK MODULUS DATA FOR WATER
C
      READ(6,1001) NWATER
 1001 FORMAT(I5)
      DO 100 I=1,NWATER
  100 READ(6,1002) PP(I),BM(I)
 1002 FORMAT(2F10.0)
C
C     READ INITIAL POROSITY: POR
C          MAXIMUM STRAIN: EMAX
C          NUMBER OF DIVISION: NDIV
C
      READ(7,1003) POR,EMAX,NDIV
 1003 FORMAT(2F10.0,I5)
C
C     SET INITIAL CONDITIONS
C
      PORI = POR
      VO = 1.0
      VWO = POR*VO
      V = VO
      VW = VWO
      STRN = 0.0
      P = 0.0
      DSTRN = EMAX/NDIV
C
C     STEP-BY-STEP CALCULATIONS
C
      DO 500 I=1,NDIV
C
C      CALCULATE GRAIN BULK MODULUS
C
      CALL BULKG(P,BKG)
C
C     CALCULATE WATER BULK MODULUS
C
      CALL BULKW(P,BKW)
C
C     CALCULATE UNDRAINED BULK MODULUS
C
      BKF = BKG*BKW/(BKW+PORI*(BKG-BKW))
C
C     CALCULATE PRESSURE INCREMENTS
C
      DP = BKF*DSTRN
C
C     UPDATE PRESSURES
C
      P = P+DP
C
C     UPDATE STRAINS AND POROSITY
C
```

```
      STRN = STRN+DSTRN
C
      DVW = PORI*VO*DP/BKW
      DVG = (1.-PORI)*VO*DP/BKG
      V = V-(DVG+DVW)
      VW = VW-DVW
      POR = VW/V
C
C     WRITE TOTAL MEAN PRESSURE: P
C           UNDRAINED BULK MODULUS: BKF
C           PORE WATER PRESSURE: P
C           POROSITY: POR
C           AS A FUNCTION OF TOTAL STRAIN: STRN
C
      WRITE(8,2001) STRN,P,BKF,POR
 2001 FORMAT(4E12.4)
C
  500 CONTINUE
C
      STOP
      END
      SUBROUTINE BULKG(P,BKG)
      IF(P.LE.1363.) BKG = 344827.
      IF(P.GT.1363.) BKG = 253.*P
      RETURN
      END
      SUBROUTINE BULKW(PF,BKW)
      COMMON /WATER/ PP(100),BM(100)
      I = 1
  100 CONTINUE
      IF(PF.GE.PP(I).AND.PF.LE.PP(I+1)) GO TO 200
      I = I+1
      GO TO 100
  200 CONTINUE
      BKW = BM(I)+(BM(I+1)-BM(I))*(PF-PP(I))/(PP(I+1)-PP(I))
      RETURN
      END
```

A.2 SAMPLE PROBLEM

A sample problem is prepared to demonstrate the computer program MENGR. TAPE6 contains the modulus of fresh water as described in Table 5.1. The initial porosity of the mixture is assumed to be 40%. Tape 8 lists the output data; p_i (bars), K_{wi} (bars), and n_i as a function of total strain, ϵ_i.

Input File Tape 6

```
   45
0.0         21837.
50.         21837.
100.        22173.
150.        22511.
200.        22854.
250.        23197.
300.        23544.
350.        23892.
400.        24245.
450.        24599.
500.        24954.
550.        25316.
600.        25675.
650.        26040.
700.        26408.
750.        26777.
800.        27149.
850.        27523.
900.        27903.
950.        28281.
1000.       28663.
1500.       31387.
2000.       35946.
3000.       42626.
4000.       51948.
5000.       61576.
6000.       71174.
7000.       81038.
8000.       91659.
9000.       1.0352E+5
10000.      1.1682E+5
15000.      1.4771E+5
20000.      1.9084E+5
25000.      2.4390E+5
30000.      2.9762E+5
40000.      3.7453E+5
50000.      4.8077E+5
60000.      5.8480E+5
70000.      6.8493E+5
80000.      7.8740E+5
90000.      8.8495E+5
1.0E+5      9.7088E+5
2.0E+5      1.4065E+6
4.0E+5      2.4038E+6
8.0E+5      6.2402E+6
```

Input File Tape 7

```
0.4         0.2             50
```

ε_i	p_i (bars)	K_{wi} (bars)	n_i
.4000E-02	.1994E+03	.4986E+05	.3979E+00
.8000E-02	.4073E+03	.5196E+05	.3959E+00
.1200E-01	.6270E+03	.5494E+05	.3938E+00
.1600E-01	.8596E+03	.5814E+05	.3918E+00
.2000E-01	.1106E+04	.6160E+05	.3897E+00
.2400E-01	.1365E+04	.6485E+05	.3877E+00
.2800E-01	.1636E+04	.6763E+05	.3856E+00
.3200E-01	.1928E+04	.7294E+05	.3835E+00
.3600E-01	.2246E+04	.7958E+05	.3814E+00
.4000E-01	.2588E+04	.8549E+05	.3792E+00
.4400E-01	.2953E+04	.9134E+05	.3769E+00
.4800E-01	.3343E+04	.9750E+05	.3747E+00
.5200E-01	.3767E+04	.1060E+06	.3723E+00
.5600E-01	.4229E+04	.1154E+06	.3700E+00
.6000E-01	.4732E+04	.1258E+06	.3676E+00
.6400E-01	.5281E+04	.1373E+06	.3652E+00
.6800E-01	.5881E+04	.1499E+06	.3628E+00
.7200E-01	.6535E+04	.1635E+06	.3603E+00
.7600E-01	.7250E+04	.1787E+06	.3578E+00
.8000E-01	.8033E+04	.1958E+06	.3553E+00
.8400E-01	.8895E+04	.2155E+06	.3528E+00
.8800E-01	.9853E+04	.2394E+06	.3502E+00
.9200E-01	.1093E+05	.2686E+06	.3476E+00
.9600E-01	.1208E+05	.2873E+06	.3450E+00
.1000E+00	.1329E+05	.3047E+06	.3424E+00
.1040E+00	.1459E+05	.3232E+06	.3397E+00
.1080E+00	.1596E+05	.3427E+06	.3370E+00
.1120E+00	.1743E+05	.3686E+06	.3342E+00
.1160E+00	.1903E+05	.3989E+06	.3315E+00
.1200E+00	.2075E+05	.4316E+06	.3287E+00
.1240E+00	.2264E+05	.4704E+06	.3258E+00
.1280E+00	.2471E+05	.5174E+06	.3230E+00
.1320E+00	.2698E+05	.5691E+06	.3201E+00
.1360E+00	.2949E+05	.6265E+06	.3172E+00
.1400E+00	.3225E+05	.6898E+06	.3143E+00
.1440E+00	.3522E+05	.7442E+06	.3114E+00
.1480E+00	.3842E+05	.7991E+06	.3084E+00
.1520E+00	.4185E+05	.8580E+06	.3054E+00
.1560E+00	.4559E+05	.9334E+06	.3023E+00
.1600E+00	.4969E+05	.1027E+07	.2993E+00
.1640E+00	.5421E+05	.1129E+07	.2962E+00
.1680E+00	.5917E+05	.1240E+07	.2931E+00
.1720E+00	.6462E+05	.1362E+07	.2899E+00
.1760E+00	.7058E+05	.1491E+07	.2867E+00
.1800E+00	.7711E+05	.1633E+07	.2835E+00
.1840E+00	.8427E+05	.1790E+07	.2803E+00
.1880E+00	.9211E+05	.1958E+07	.2770E+00
.1920E+00	.1006E+06	.2134E+07	.2737E+00
.1960E+00	.1099E+06	.2302E+07	.2703E+00
.2000E+00	.1195E+06	.2403E+07	.2670E+00

List of Output File Tape 8

APPENDIX B

UNDRAINED BEHAVIOR OF MIXTURE MODEL BASED ON TRUE STRAIN

The computer program, MTRUE, listed in subsection B.1 uses two input files: TAPE6 and TAPE7. The descriptions of input data in TAPE6 and TAPE7 are identical to those presented in Appendix A.

The same sample problem as in Appendix A.2 is solved using the Program MTRUE. Subsection B.2 lists the output file, TAPE8, which contains the calculations of: p_i (bars), $\bar{K}_{wi}$, (bars), and n_i as a function of total strain, ϵ_i.

B.1 List of Program MTRUE

```
      PROGRAM MTRUE
      COMMON /WATER/ PP(100),BM(100)
C
C     READ PRESSURE-BULK MODULUS DATA FOR WATER
C
      READ(6,1001) NWATER
 1001 FORMAT(I5)
      DO 100 I=1,NWATER
  100 READ(6,1002) PP(I),BM(I)
 1002 FORMAT(2F10.0)
      EO = 0.0
      EBO = 0.0
      VALOG = ALOG(1.-0.8)
      WALOG = ALOG10(100.)
C
      DO 105 I=2,NWATER
      E = EO+(PP(I)-PP(I-1))/BM(I-1)
      EB = -ALOG(1.-E)
      BM(I-1) = (PP(I)-PP(I-1))/(EB-EBO)
      EO = E
      EBO = EB
  105 CONTINUE
C
C     READ INITIAL POROSITY: POR
C          MAXIMUM STRAIN: EMAX
C          NUMBER OF DIVISION: NDIV
C
      READ(7,1003) POR,EMAX,NDIV
 1003 FORMAT(2F10.0,I5)
C
C     SET INITIAL CONDITIONS
C
      STRN = 0.0
      P = 0.0
      DSTRN = EMAX/NDIV
C
C     STEP-BY-STEP CALCULATIONS
C
      DO 500 I=1,NDIV
C
      SBRNO = -ALOG(1.-STRN)
      SBRN = -ALOG(1.-STRN-DSTRN)
      DSBRN = SBRN-SBRNO
C
C      CALCULATE GRAIN BULK MODULUS
C
      CALL BULKG(P,BKG)
C
C     CALCULATE WATER BULK MODULUS
C
      CALL BULKW(P,BKW)
C
C     CALCULATE UNDRAINED BULK MODULUS
C
      BKF = BKG*BKW/(BKW+POR*(BKG-BKW))
```

```
C
C     CALCULATE PRESSURE INCREMENTS
C
      DP = BKF*DSBRN
C
C     UPDATE PRESSURES
C
      P = P+DP
C
C     UPDATE STRAINS AND POROSITY
C
      STRN = STRN+DSTRN
C
      POR = POR*(1.-DP/BKW)/(1.-DSBRN)
C
C     WRITE TOTAL MEAN PRESSURE: P
C           UNDRAINED BULK MODULUS: BKF
C           PORE WATER PRESSURE: P
C           POROSITY: POR
C           AS A FUNCTION OF TOTAL STRAIN: STRN
C
      WRITE(8,2001) STRN,P,BKF,POR
 2001 FORMAT(4E12.4)
C
  500 CONTINUE
C
      STOP
      END
      SUBROUTINE BULKG(P,BKG)
      IF(P.LE.1369.5) BKG = 344827.
      IF(P.GT.1369.5) BKG=253.*P*(1.-(ALOG10(P/406.9))/109.9)
      RETURN
      END
      SUBROUTINE BULKW(PF,BKW)
      COMMON /WATER/ PP(100),BM(100)
      I = 1
  100 CONTINUE
      IF(PF.GE.PP(I).AND.PF.LE.PP(I+1)) GO TO 200
      I = I+1
      GO TO 100
  200 CONTINUE
      BKW = BM(I)+(BM(I+1)-BM(I))*(PF-PP(I))/(PP(I+1)-PP(I))
      RETURN
      END
```

ε_i	p_i (bars)	$\overline{K}_{wi}$ (bars)	n_i
.4000E-02	.1996E+03	.4987E+05	.3979E+00
.8000E-02	.4079E+03	.5169E+05	.3959E+00
.1200E-01	.6280E+03	.5452E+05	.3938E+00
.1600E-01	.8613E+03	.5748E+05	.3917E+00
.2000E-01	.1108E+04	.6059E+05	.3897E+00
.2400E-01	.1366E+04	.6314E+05	.3876E+00
.2800E-01	.1635E+04	.6556E+05	.3856E+00
.3200E-01	.1925E+04	.7024E+05	.3835E+00
.3600E-01	.2240E+04	.7599E+05	.3813E+00
.4000E-01	.2577E+04	.8117E+05	.3791E+00
.4400E-01	.2938E+04	.8631E+05	.3769E+00
.4800E-01	.3322E+04	.9174E+05	.3746E+00
.5200E-01	.3740E+04	.9908E+05	.3722E+00
.5600E-01	.4194E+04	.1074E+06	.3699E+00
.6000E-01	.4689E+04	.1165E+06	.3675E+00
.6400E-01	.5229E+04	.1266E+06	.3651E+00
.6800E-01	.5817E+04	.1375E+06	.3626E+00
.7200E-01	.6460E+04	.1494E+06	.3602E+00
.7600E-01	.7162E+04	.1624E+06	.3577E+00
.8000E-01	.7929E+04	.1770E+06	.3551E+00
.8400E-01	.8774E+04	.1938E+06	.3526E+00
.8800E-01	.9712E+04	.2141E+06	.3500E+00
.9200E-01	.1075E+05	.2352E+06	.3474E+00
.9600E-01	.1185E+05	.2507E+06	.3448E+00
.1000E+00	.1302E+05	.2642E+06	.3422E+00
.1040E+00	.1426E+05	.2786E+06	.3395E+00
.1080E+00	.1558E+05	.2939E+06	.3367E+00
.1120E+00	.1698E+05	.3126E+06	.3340E+00
.1160E+00	.1850E+05	.3360E+06	.3312E+00
.1200E+00	.2014E+05	.3614E+06	.3284E+00
.1240E+00	.2191E+05	.3895E+06	.3256E+00
.1280E+00	.2386E+05	.4254E+06	.3227E+00
.1320E+00	.2600E+05	.4650E+06	.3198E+00
.1360E+00	.2834E+05	.5072E+06	.3169E+00
.1400E+00	.3090E+05	.5515E+06	.3140E+00
.1440E+00	.3368E+05	.5958E+06	.3110E+00
.1480E+00	.3665E+05	.6350E+06	.3080E+00
.1520E+00	.3984E+05	.6773E+06	.3050E+00
.1560E+00	.4326E+05	.7231E+06	.3019E+00
.1600E+00	.4700E+05	.7880E+06	.2988E+00
.1640E+00	.5111E+05	.8607E+06	.2957E+00
.1680E+00	.5562E+05	.9401E+06	.2926E+00
.1720E+00	.6056E+05	.1025E+07	.2894E+00
.1760E+00	.6597E+05	.1117E+07	.2862E+00
.1800E+00	.7188E+05	.1214E+07	.2830E+00
.1840E+00	.7834E+05	.1321E+07	.2797E+00
.1880E+00	.8541E+05	.1439E+07	.2764E+00
.1920E+00	.9312E+05	.1562E+07	.2731E+00
.1960E+00	.1013E+06	.1641E+07	.2697E+00
.2000E+00	.1095E+06	.1643E+07	.2663E+00

B.2 List of Sample Problem Output

APPENDIX C

UNDRAINED BEHAVIOR OF FULLY COUPLED MODEL IN HYDROSTATIC COMPRESSION

The computer program ISOCP, listed in subsection C.1, was input from TAPE5 and TAPE7. TAPE7 contains the water modulus data described in Table 5.1. TAPE5 contains the initial porosity and the skeleton effective mean pressure vs. effective volume strain response. The compressibility of the solid grains from Section 5.2.2 is built into subroutine BULKG in the tabular form.

DESCRIPTION OF INPUT FILES

TAPE7 - See Appendix A (TAPE6)

TAPE5

Card 1

POR, NPOINT, NDIV, EMAX (free format)

POR = Initial porositiy
NPOINT = Number of effective volume strain/skeleton bulk modulus pairs
NDIV = Total number of steps
EMAX = Maximum total strain at which calculation terminates

Card 2

$$\left.\begin{array}{l}\epsilon'_1, p'_1\\ \epsilon'_2, p'_2\\ \epsilon'_n, p'_n\end{array}\right\}$$

NPOINT cards with free format for each card.
Note that p'_i is in terms of bars.

The output is contained in TAPE6. TAPE6 lists total mean pressure (p_i), undrained bulk modulus (K_{fi}), effective mean pressure (p'_i), pore water pressure (π_i), and current porosity (n_i) as functions of total volume strain (ϵ_i).

C.1 List of Program ISOCP

```
FTN7X
$FILES(0.3)
      PROGRAM ISOCP
C
C REVISION: October 10, 1985
C REVISION: October 11, 1985
C REVISION: October 23, 1985
C REVISION: February 28, 1986
C REVISION: July 28, 1986 (changed lines 95 and 96)
C
      COMMON/WATER/PP(300),BM(300)
      COMMON/SKTON/RR(300),CM(300)
      INTEGER NAM1(6),NAM2(6),NAM3(6)
C
C Read bulk modulus of water as a function of pressure.
C
      WRITE(1,*)'    '
      WRITE(1,*)'The PRESSURE-BULK MODULUS file has # of data pairs'
      WRITE(1,*)'    at the top with pairs in bars '
      WRITE(1,*)'The STRAIN-STRESS file has INITIAL POROSITY, # of    '
      WRITE(1,*)'    data pairs, # of divi., and max strain at top,'
      WRITE(1,*)'    with pairs of STRAIN not in % and STRESS (bars)'
      WRITE(1,*)'    '
      WRITE(1,*)'**NOTE: STRAIN-STRESS file must begin with zero**'
      WRITE(1,*)'      '
      WRITE(1)'Type INPUT file with PRESSURE-BULK MODULUS pairs  '
      READ(1,10) NAM1
10    FORMAT(6A2)
      WRITE(1)'Type INPUT file with STRAIN-STRESS pairs          '
      READ(1,10) NAM2
      OPEN(7,FILE=NAM1)
      WRITE(1)'Type OUTPUT file                                  '
      READ(1,10) NAM3
      WRITE(1,*)'                '
      WRITE(1)'Type case A.B or C                                '
      READ(1,30) IANS
30    FORMAT(A1)
C
      OPEN(6.FILE=NAM3)
      READ(7.*) NWATER
      DO 100 I=1.NWATER
100   READ(7.*) PP(I).BM(I)
      CLOSE(7)
C
C Read initial porosity and # of data pairs in STRAIN-STRESS curve
C       and # of divisions and maximum strain to calculate
C
      OPEN(5.FILE=NAM2)
      READ(5,*) POR,NPOINT,NDIV.EMAX
      DO 120 I=1.NPOINT
120   READ(5,*)RR(I),CM(I)
      CLOSE(5)
C
```

```
C Set initial conditions
C
      VO = 1.0
      PORI = POR
      STRN = 0.0
      EE = 0.0
      P = 0.0
      PE = 0.0
      PF = 0.0
      NFIRST = 0
      KK = 1
C
      OPEN(6,FILE=NAM3)
C
C Steo-by-steo calculations
C
 150  IF(NFIRST.EQ.0) GOTO 200
      K = K + 1
      IF(K.EQ.NDIV) GOTO 200
      GOTO 300
200   CONTINUE
      K = 0
      NFIRST = 1
      KK = KK + 1
      DDTRN = (RR(KK)-RR(KK-1))/NDIV
300   CONTINUE
C
C Calculate skeleton bulk modulus
C
            CALL CONMS(EE,BKS)
C
C Calculate grain bulk modulus
C
            CALL BULKG(PF,PE,POR,BKG,IANS)
C
C Calculate water bulk modulus
C
            CALL BULKW(PF,BKW)
C
C Calculate undrained bulk modulus
C
            AA = (1.-PORI)/(1.-POR)
            BKF = BKS+(BKG-AA*BKS)/(1.+PORI*BKG*(BKG-BKW)/
     #             (BKW*(BKG-AA*BKS)))
C
C Calculate total strain increment
C
            DSTRN = DDTRN*(BKG-BKS)/(BKG-BKF)
C
C Calculate oressure increments
C
            DP = BKF*DSTRN
            DPF = DSTRN*(BKF-BKS)/(1.-BKS/BKG)
            DPE = DP-DPF
C
C Uodate oressures
C
            P = P+DP
            PF = PF+DPF
            PE = PE+DPE
C
```

```
C Update strain, volume of water and porosity
C
            EE = EE+DDTRN
            STRNO = STRN
            PORO = POR
            STRN = STRN +DSTRN
            DVW = PORO*(1.-STRNO)-PORI*DPF/BKW
            POR = DVW/(1.-STRN)
C
C Write:
C           Total mean pressure (bars), (P)
C           Undrained bulk modulus (bars), (BKF)
C           Effective mean pressure (bars), (PE)
C           Pore water pressure (PF)
C           and porosity (POR)
C           as a function of strain (%), (STRN)
C
            WRITE(6,2001) STRN*100,P,BKF,PE,PF,POR
2001        FORMAT(6E12.4)
            IF(STRN.GT.EMAX) GOTO 500
            GOTO 150
 500  CONTINUE
      CLOSE(6)
      STOP
      END
C
C*************************************************
C
      SUBROUTINE BULKG(PF,PE,POR,BKG,IANS)
C
C Assume constant
C
      IF(IANS.NE.1HA) THEN
      P = PF+PE/(1.-POR)
      IF(P.LE.1363.) BKG = 344827.
      IF(P.GT.1363.) BKG = 253. * P
      ENDIF
      IF(IANS.EQ.1HB) THEN
      IF(BKG.GT.688000.) BKG = 688000.
      ENDIF
      IF(IANS.EQ.1HA)  BKG = 413793.  !In bars
      RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE BULKW(PF,BKW)
C
      COMMON/WATER/PP(300),BM(300)
C
      I = 1
100   CONTINUE
      IF(PF.GE.PP(I).AND.PF.LE.PP(I+1)) GOTO 200
      I = I+1
      GOTO 100
200   CONTINUE
      BKW = BM(I)+(BM(I+1)-BM(I))*(PF-PP(I))/(PP(I+1)-PP(I))
      RETURN
      END
```

```
C
      SUBROUTINE CONMS(EE,BMS)
      COMMON/SKTON/RR(300),CM(300)
      I = 1
100   CONTINUE
      IF(EE.GE.RR(I).AND.EE.LE.RR(I+1)) GOTO 200
      I = I +1
      GOTO 100
200   CONTINUE
      BMS = (CM(I+1)-CM(I))/(RR(I+1)-RR(I))
      RETURN
      END
C
```

C.2 SAMPLE PROBLEMS

Two sample problems were prepared to illustrate the use of ISOCP in uncemented and cemented porous media.

C.2.1 Beach Sand

TAPE5 contains the input properties of the uncemented beach sand skeleton expressed in a list of effective volume strain-effective mean pressure pairs.

TAPE5

```
0.4        15        50        .200

0.0        0.0
 .034       91.65
 .062      167.13
 .091      245.3
 .115      310.
 .161      517.
 .187      676.
 .216      976.
 .235     1228.
 .249     1503.
 .264     1862.
 .277     2324.
 .288     2848.
 .298     3503.
 .301     3793.
```

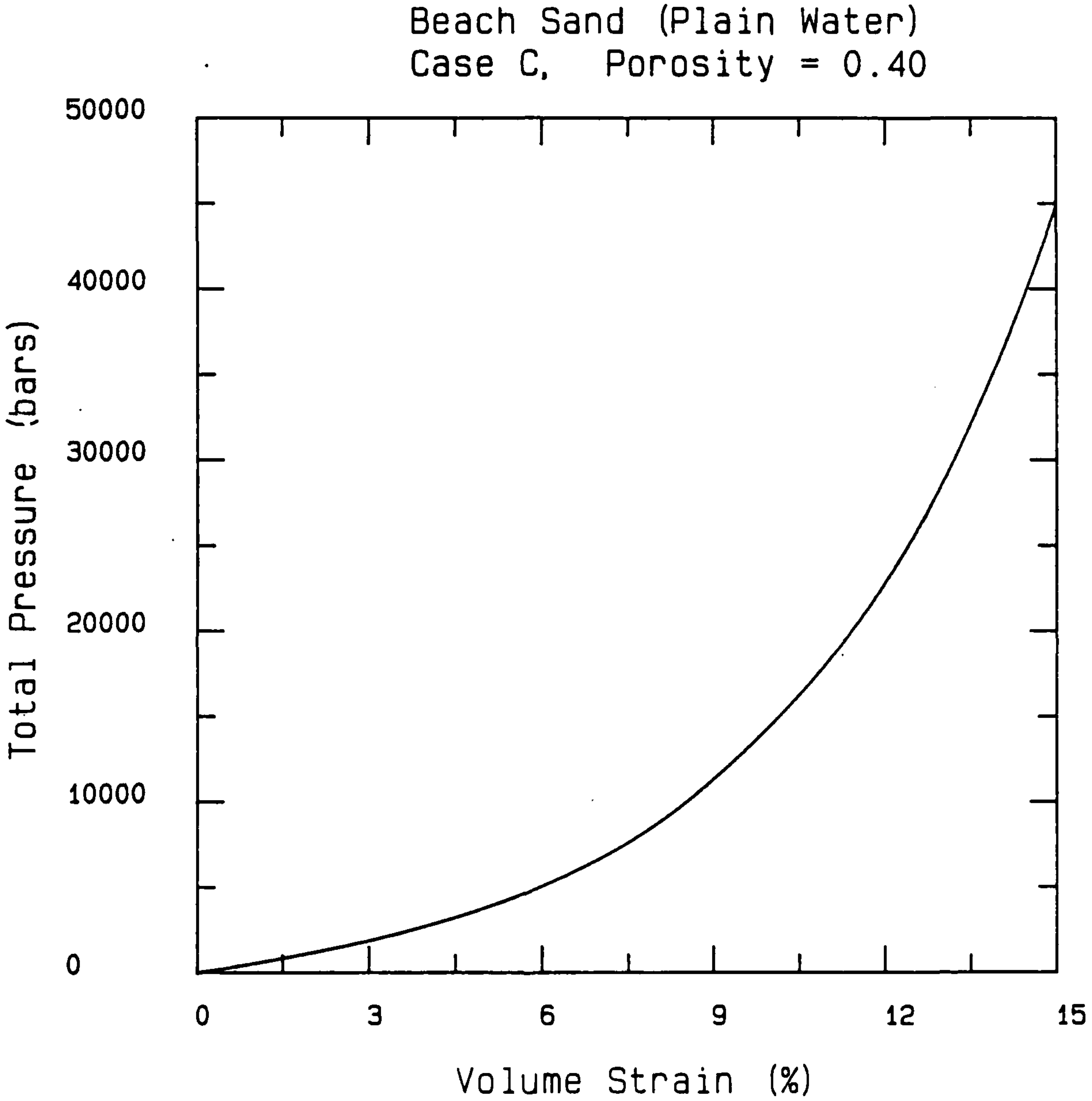
TAPE6 lists the output data; ϵ_i (%), p_i (bars), K_{fi} (bars), p'_i (bars), π_i (bars), and n_i.

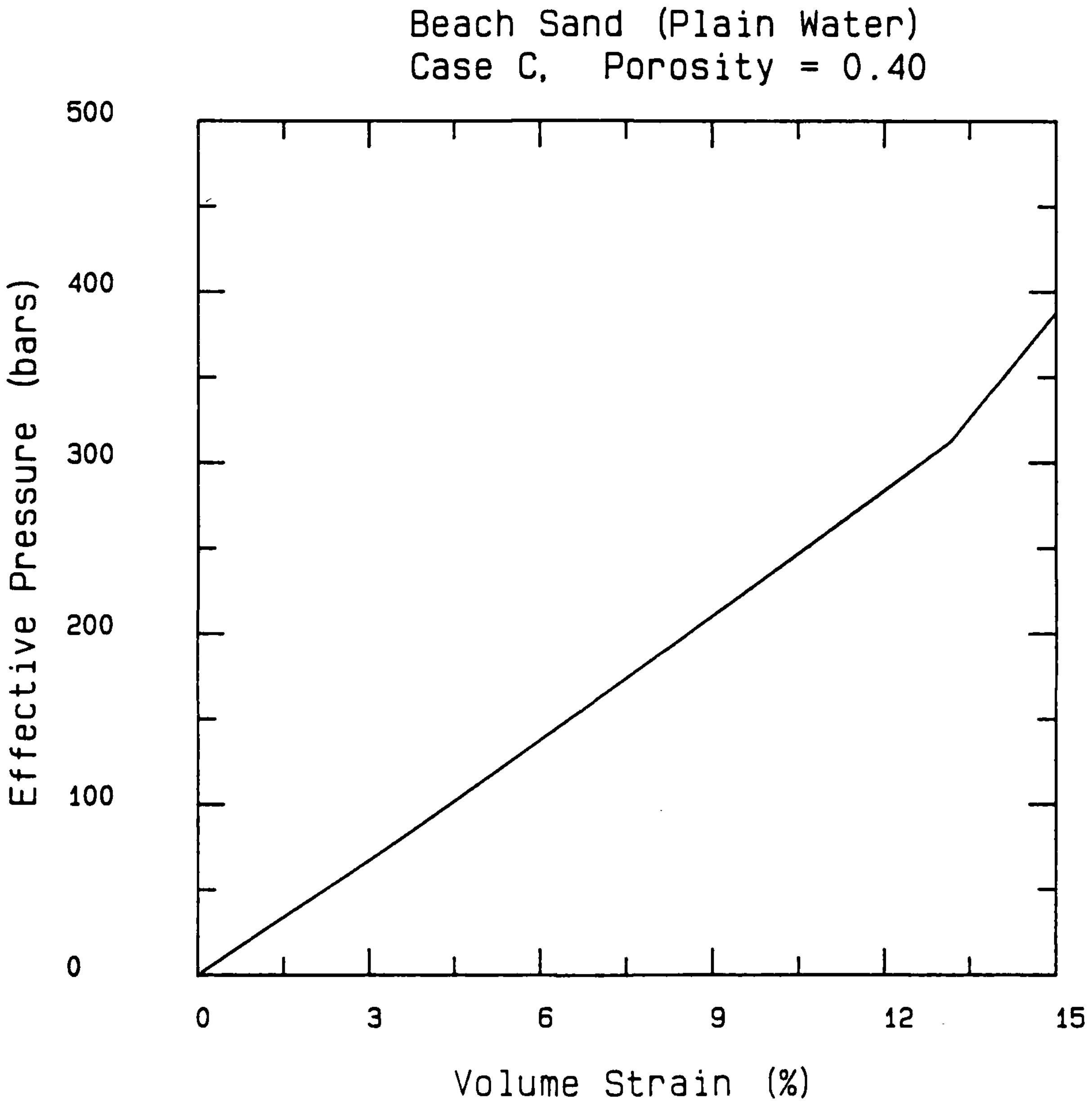
List of Output File Tape 6

ε_i (%)	p_i (bars)	K_{fi} (bars)	p'_i (bars)	π_i (bars)	n_i
.1588E+00	.8231E+02	.5183E+05	.3666E+01	.7865E+02	.3992E+00
.3178E+00	.1654E+03	.5223E+05	.7332E+01	.1580E+03	.3984E+00
.4775E+00	.2505E+03	.5333E+05	.1100E+02	.2395E+03	.3976E+00
.6377E+00	.3378E+03	.5447E+05	.1466E+02	.3231E+03	.3968E+00
.7986E+00	.4273E+03	.5565E+05	.1833E+02	.4090E+03	.3959E+00
.9602E+00	.5192E+03	.5687E+05	.2200E+02	.4972E+03	.3951E+00
.1122E+01	.6136E+03	.5813E+05	.2566E+02	.5879E+03	.3943E+00
.1286E+01	.7105E+03	.5944E+05	.2933E+02	.6812E+03	.3935E+00
.1449E+01	.8101E+03	.6080E+05	.3299E+02	.7771E+03	.3926E+00
.1614E+01	.9125E+03	.6221E+05	.3666E+02	.8759E+03	.3918E+00
.1779E+01	.1018E+04	.6367E+05	.4033E+02	.9776E+03	.3910E+00
.1946E+01	.1126E+04	.6519E+05	.4399E+02	.1082E+04	.3901E+00
.2113E+01	.1237E+04	.6640E+05	.4766E+02	.1190E+04	.3893E+00
.2281E+01	.1351E+04	.6753E+05	.5132E+02	.1299E+04	.3885E+00
.2449E+01	.1466E+04	.6881E+05	.5499E+02	.1411E+04	.3876E+00
.2615E+01	.1584E+04	.7068E+05	.5866E+02	.1525E+04	.3868E+00
.2779E+01	.1703E+04	.7271E+05	.6232E+02	.1641E+04	.3859E+00
.2942E+01	.1826E+04	.7539E+05	.6599E+02	.1760E+04	.3851E+00
.3104E+01	.1952E+04	.7814E+05	.6965E+02	.1883E+04	.3842E+00
.3265E+01	.2083E+04	.8095E+05	.7332E+02	.2010E+04	.3833E+00
.3425E+01	.2217E+04	.8378E+05	.7699E+02	.2140E+04	.3825E+00
.3585E+01	.2354E+04	.8607E+05	.8065E+02	.2274E+04	.3816E+00
.3743E+01	.2494E+04	.8839E+05	.8432E+02	.2410E+04	.3807E+00
.3901E+01	.2637E+04	.9075E+05	.8798E+02	.2549E+04	.3798E+00
.4058E+01	.2783E+04	.9313E+05	.9165E+02	.2692E+04	.3790E+00
.4186E+01	.2906E+04	.9556E+05	.9467E+02	.2812E+04	.3782E+00
.4315E+01	.3032E+04	.9759E+05	.9769E+02	.2934E+04	.3775E+00
.4443E+01	.3159E+04	.9966E+05	.1007E+03	.3058E+04	.3768E+00
.4570E+01	.3289E+04	.1021E+06	.1037E+03	.3186E+04	.3761E+00
.4698E+01	.3423E+04	.1049E+06	.1067E+03	.3316E+04	.3753E+00
.4825E+01	.3560E+04	.1079E+06	.1098E+03	.3451E+04	.3746E+00
.4952E+01	.3701E+04	.1109E+06	.1128E+03	.3588E+04	.3738E+00
.5079E+01	.3846E+04	.1139E+06	.1158E+03	.3730E+04	.3731E+00
.5205E+01	.3994E+04	.1171E+06	.1188E+03	.3875E+04	.3724E+00
.5332E+01	.4146E+04	.1203E+06	.1218E+03	.4024E+04	.3716E+00
.5458E+01	.4302E+04	.1237E+06	.1249E+03	.4176E+04	.3709E+00
.5584E+01	.4463E+04	.1272E+06	.1279E+03	.4335E+04	.3701E+00
.5711E+01	.4628E+04	.1308E+06	.1309E+03	.4497E+04	.3694E+00
.5837E+01	.4797E+04	.1345E+06	.1339E+03	.4664E+04	.3686E+00
.5962E+01	.4972E+04	.1383E+06	.1369E+03	.4835E+04	.3679E+00
.6088E+01	.5151E+04	.1423E+06	.1400E+03	.5011E+04	.3671E+00
.6214E+01	.5334E+04	.1463E+06	.1430E+03	.5191E+04	.3664E+00
.6339E+01	.5523E+04	.1504E+06	.1460E+03	.5377E+04	.3656E+00
.6465E+01	.5717E+04	.1547E+06	.1490E+03	.5568E+04	.3648E+00

```
.6590E+01    .5917E+04    .1590E+06    .1520E+03    .5765E+04    .3641E+00
.6715E+01    .6121E+04    .1635E+06    .1551E+03    .5968E+04    .3632E+00
.6841E+01    .6332E+04    .1681E+06    .1581E+03    .6174E+04    .3625E+00
.6966E+01    .6548E+04    .1729E+06    .1611E+03    .6387E+04    .3616E+00
.7091E+01    .6771E+04    .1779E+06    .1641E+03    .6607E+04    .3610E+00
.7216E+01    .7000E+04    .1830E+06    .1671E+03    .6833E+04    .3602E+00
.7345E+01    .7243E+04    .1883E+06    .1703E+03    .7073E+04    .3594E+00
.7475E+01    .7494E+04    .1941E+06    .1734E+03    .7321E+04    .3586E+00
.7604E+01    .7753E+04    .2003E+06    .1765E+03    .7577E+04    .3578E+00
.7733E+01    .8020E+04    .2067E+06    .1796E+03    .7841E+04    .3570E+00
.7863E+01    .8296E+04    .2133E+06    .1828E+03    .8113E+04    .3562E+00
.7992E+01    .8581E+04    .2204E+06    .1859E+03    .8395E+04    .3554E+00
.8121E+01    .8876E+04    .2282E+06    .1890E+03    .8687E+04    .3545E+00
.8250E+01    .9182E+04    .2363E+06    .1921E+03    .8990E+04    .3537E+00
.8380E+01    .9498E+04    .2447E+06    .1953E+03    .9303E+04    .3529E+00
.8509E+01    .9827E+04    .2543E+06    .1984E+03    .9629E+04    .3521E+00
.8639E+01    .1017E+05    .2644E+06    .2015E+03    .9968E+04    .3512E+00
.8768E+01    .1053E+05    .2749E+06    .2047E+03    .1032E+05    .3504E+00
.8898E+01    .1089E+05    .2808E+06    .2078E+03    .1068E+05    .3496E+00
.9027E+01    .1126E+05    .2863E+06    .2109E+03    .1105E+05    .3487E+00
.9156E+01    .1164E+05    .2918E+06    .2140E+03    .1142E+05    .3479E+00
.9285E+01    .1202E+05    .2975E+06    .2172E+03    .1180E+05    .3471E+00
.9413E+01    .1241E+05    .3033E+06    .2203E+03    .1219E+05    .3462E+00
.9542E+01    .1281E+05    .3092E+06    .2234E+03    .1258E+05    .3454E+00
.9670E+01    .1321E+05    .3151E+06    .2265E+03    .1298E+05    .3445E+00
.9798E+01    .1362E+05    .3212E+06    .2297E+03    .1339E+05    .3437E+00
.9926E+01    .1404E+05    .3274E+06    .2328E+03    .1381E+05    .3428E+00
.1005E+02    .1447E+05    .3336E+06    .2359E+03    .1423E+05    .3420E+00
.1018E+02    .1490E+05    .3400E+06    .2390E+03    .1466E+05    .3411E+00
.1031E+02    .1534E+05    .3465E+06    .2422E+03    .1510E+05    .3402E+00
.1044E+02    .1579E+05    .3537E+06    .2453E+03    .1555E+05    .3394E+00
.1054E+02    .1618E+05    .3629E+06    .2479E+03    .1593E+05    .3387E+00
.1065E+02    .1657E+05    .3707E+06    .2505E+03    .1632E+05    .3380E+00
.1075E+02    .1697E+05    .3787E+06    .2531E+03    .1671E+05    .3372E+00
.1086E+02    .1737E+05    .3868E+06    .2557E+03    .1712E+05    .3365E+00
.1096E+02    .1779E+05    .3951E+06    .2582E+03    .1753E+05    .3358E+00
.1107E+02    .1821E+05    .4036E+06    .2608E+03    .1795E+05    .3351E+00
.1117E+02    .1865E+05    .4123E+06    .2634E+03    .1839E+05    .3343E+00
.1128E+02    .1909E+05    .4212E+06    .2660E+03    .1883E+05    .3336E+00
.1138E+02    .1955E+05    .4302E+06    .2686E+03    .1928E+05    .3329E+00
.1149E+02    .2001E+05    .4395E+06    .2712E+03    .1974E+05    .3322E+00
.1159E+02    .2048E+05    .4489E+06    .2738E+03    .2021E+05    .3314E+00
.1170E+02    .2096E+05    .4595E+06    .2764E+03    .2069E+05    .3307E+00
.1181E+02    .2146E+05    .4715E+06    .2789E+03    .2119E+05    .3300E+00
.1191E+02    .2197E+05    .4838E+06    .2815E+03    .2169E+05    .3292E+00
.1202E+02    .2249E+05    .4965E+06    .2841E+03    .2221E+05    .3285E+00
.1212E+02    .2303E+05    .5095E+06    .2867E+03    .2274E+05    .3277E+00
.1223E+02    .2358E+05    .5228E+06    .2893E+03    .2329E+05    .3270E+00
.1233E+02    .2414E+05    .5365E+06    .2919E+03    .2385E+05    .3263E+00
.1244E+02    .2473E+05    .5506E+06    .2945E+03    .2443E+05    .3255E+00
.1254E+02    .2532E+05    .5650E+06    .2971E+03    .2502E+05    .3248E+00
.1265E+02    .2593E+05    .5798E+06    .2996E+03    .2563E+05    .3240E+00
```

```
.1275E+02   .2656E+05   .5952E+06   .3022E+03   .2626E+05   .3233E+00
.1286E+02   .2720E+05   .6110E+06   .3048E+03   .2690E+05   .3225E+00
.1296E+02   .2787E+05   .6273E+06   .3074E+03   .2756E+05   .3218E+00
.1307E+02   .2855E+05   .6439E+06   .3100E+03   .2824E+05   .3210E+00
.1327E+02   .2988E+05   .6610E+06   .3150E+03   .2957E+05   .3195E+00
.1348E+02   .3129E+05   .6962E+06   .3232E+03   .3097E+05   .3181E+00
.1368E+02   .3276E+05   .7250E+06   .3315E+03   .3243E+05   .3166E+00
.1388E+02   .3428E+05   .7520E+06   .3398E+03   .3394E+05   .3151E+00
.1408E+02   .3586E+05   .7798E+06   .3481E+03   .3551E+05   .3136E+00
.1428E+02   .3749E+05   .8087E+06   .3564E+03   .3713E+05   .3122E+00
.1449E+02   .3918E+05   .8387E+06   .3646E+03   .3881E+05   .3107E+00
.1469E+02   .4093E+05   .8697E+06   .3729E+03   .4056E+05   .3092E+00
.1489E+02   .4276E+05   .9055E+06   .3812E+03   .4238E+05   .3076E+00
.1509E+02   .4467E+05   .9510E+06   .3895E+03   .4428E+05   .3061E+00
.1529E+02   .4669E+05   .9987E+06   .3978E+03   .4629E+05   .3046E+00
.1549E+02   .4880E+05   .1049E+07   .4060E+03   .4840E+05   .3031E+00
.1570E+02   .5103E+05   .1102E+07   .4143E+03   .5061E+05   .3015E+00
.1590E+02   .5336E+05   .1157E+07   .4226E+03   .5294E+05   .3000E+00
.1610E+02   .5582E+05   .1214E+07   .4309E+03   .5539E+05   .2984E+00
.1630E+02   .5839E+05   .1274E+07   .4392E+03   .5795E+05   .2969E+00
.1650E+02   .6109E+05   .1337E+07   .4474E+03   .6065E+05   .2953E+00
.1671E+02   .6393E+05   .1402E+07   .4557E+03   .6347E+05   .2937E+00
.1691E+02   .6690E+05   .1469E+07   .4640E+03   .6644E+05   .2921E+00
.1711E+02   .7001E+05   .1539E+07   .4723E+03   .6954E+05   .2905E+00
.1731E+02   .7327E+05   .1612E+07   .4806E+03   .7279E+05   .2889E+00
.1751E+02   .7670E+05   .1691E+07   .4888E+03   .7621E+05   .2873E+00
.1772E+02   .8028E+05   .1773E+07   .4971E+03   .7979E+05   .2857E+00
.1792E+02   .8405E+05   .1860E+07   .5054E+03   .8354E+05   .2841E+00
.1812E+02   .8799E+05   .1946E+07   .5137E+03   .8748E+05   .2824E+00
.1824E+02   .9032E+05   .2037E+07   .5184E+03   .8980E+05   .2815E+00
.1835E+02   .9271E+05   .2092E+07   .5247E+03   .9219E+05   .2806E+00
.1847E+02   .9516E+05   .2141E+07   .5311E+03   .9463E+05   .2796E+00
.1858E+02   .9767E+05   .2191E+07   .5374E+03   .9713E+05   .2787E+00
.1869E+02   .1002E+06   .2243E+07   .5438E+03   .9969E+05   .2778E+00
.1881E+02   .1029E+06   .2295E+07   .5502E+03   .1023E+06   .2768E+00
.1892E+02   .1055E+06   .2327E+07   .5565E+03   .1050E+06   .2759E+00
.1904E+02   .1082E+06   .2356E+07   .5629E+03   .1076E+06   .2749E+00
.1915E+02   .1109E+06   .2385E+07   .5692E+03   .1103E+06   .2740E+00
.1926E+02   .1137E+06   .2415E+07   .5756E+03   .1131E+06   .2730E+00
.1938E+02   .1164E+06   .2444E+07   .5820E+03   .1159E+06   .2721E+00
.1949E+02   .1192E+06   .2475E+07   .5883E+03   .1187E+06   .2711E+00
.1960E+02   .1221E+06   .2505E+07   .5947E+03   .1215E+06   .2701E+00
.1972E+02   .1250E+06   .2536E+07   .6010E+03   .1244E+06   .2692E+00
.1983E+02   .1279E+06   .2567E+07   .6074E+03   .1273E+06   .2681E+00
.1994E+02   .1308E+06   .2599E+07   .6138E+03   .1302E+06   .2673E+00
.2006E+02   .1338E+06   .2630E+07   .6201E+03   [illegible]   [illegible]
```


Beach Sand (Plain Water)
Case C, Porosity = 0.40
50000
40000
30000
20000
10000
0
Total Pressure (bars)
0
3
6
9
12
15
Volume Strain (%)


Beach Sand (Plain Water)
Case C, Porosity = 0.40
Effective Pressure (bars)
500
400
300
200
100
0
0
3
6
9
12
15
Volume Strain (%)

C.2.2 Cemented Coral

TAPE5 contains the input properties of the cemented coral skeleton expressed in a table of effective volume strain-effective mean pressure pairs.

TAPE5

```
0.41        13        50        .200

0.0           0.0
3.48E-3     255.96
4.E-2       288.35
7.6E-2      337.33
 .132       415.54
 .18        496.12
 .206       569.59
 .219       616.99
 .241       724.43
 .262       854.78
 .276       991.45
 .284      1079.1
 .287      1132.9
```

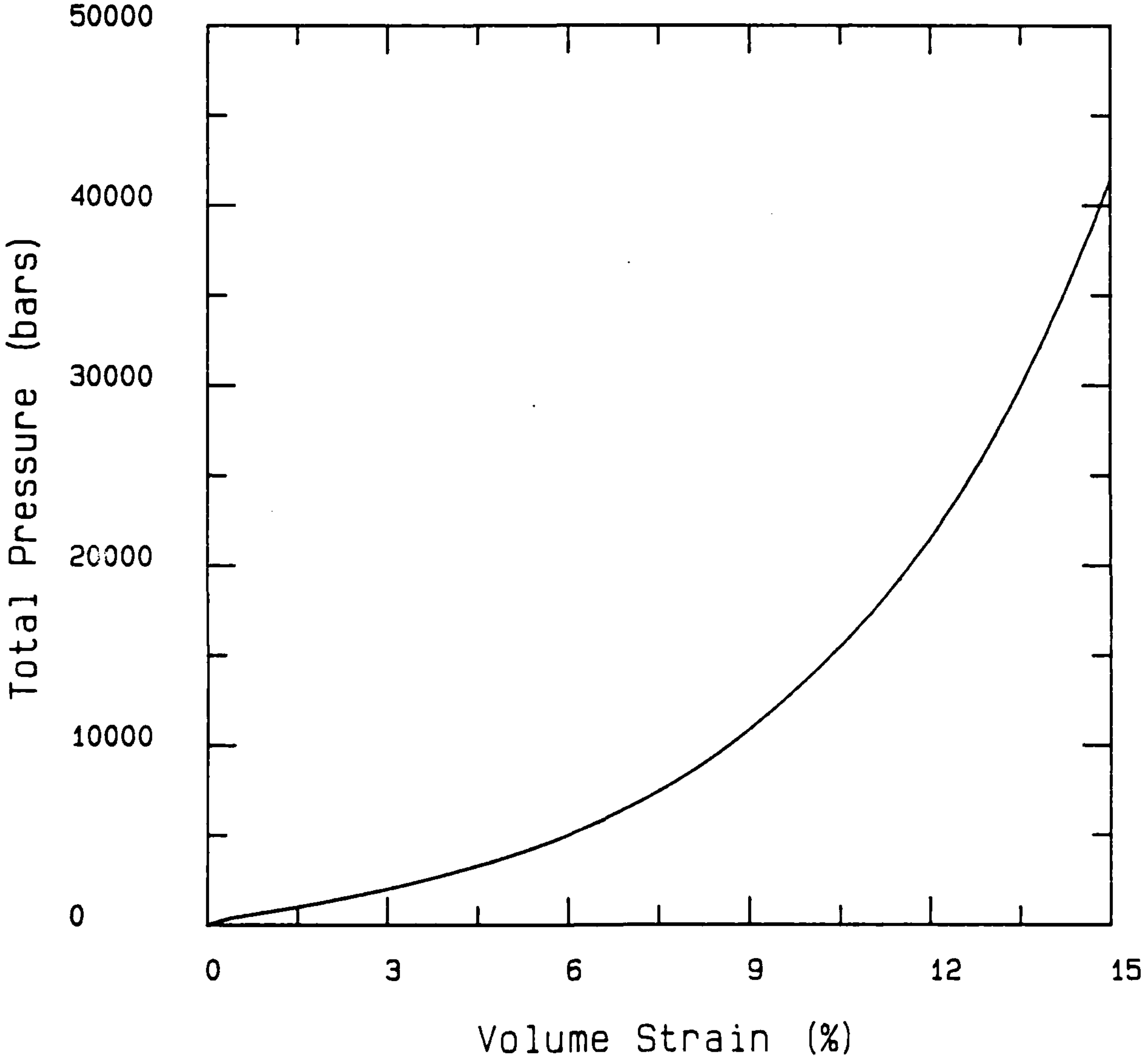
TAPE6 lists the output data in the same format as described in Section C.2.1.

List of Output File Tape 6

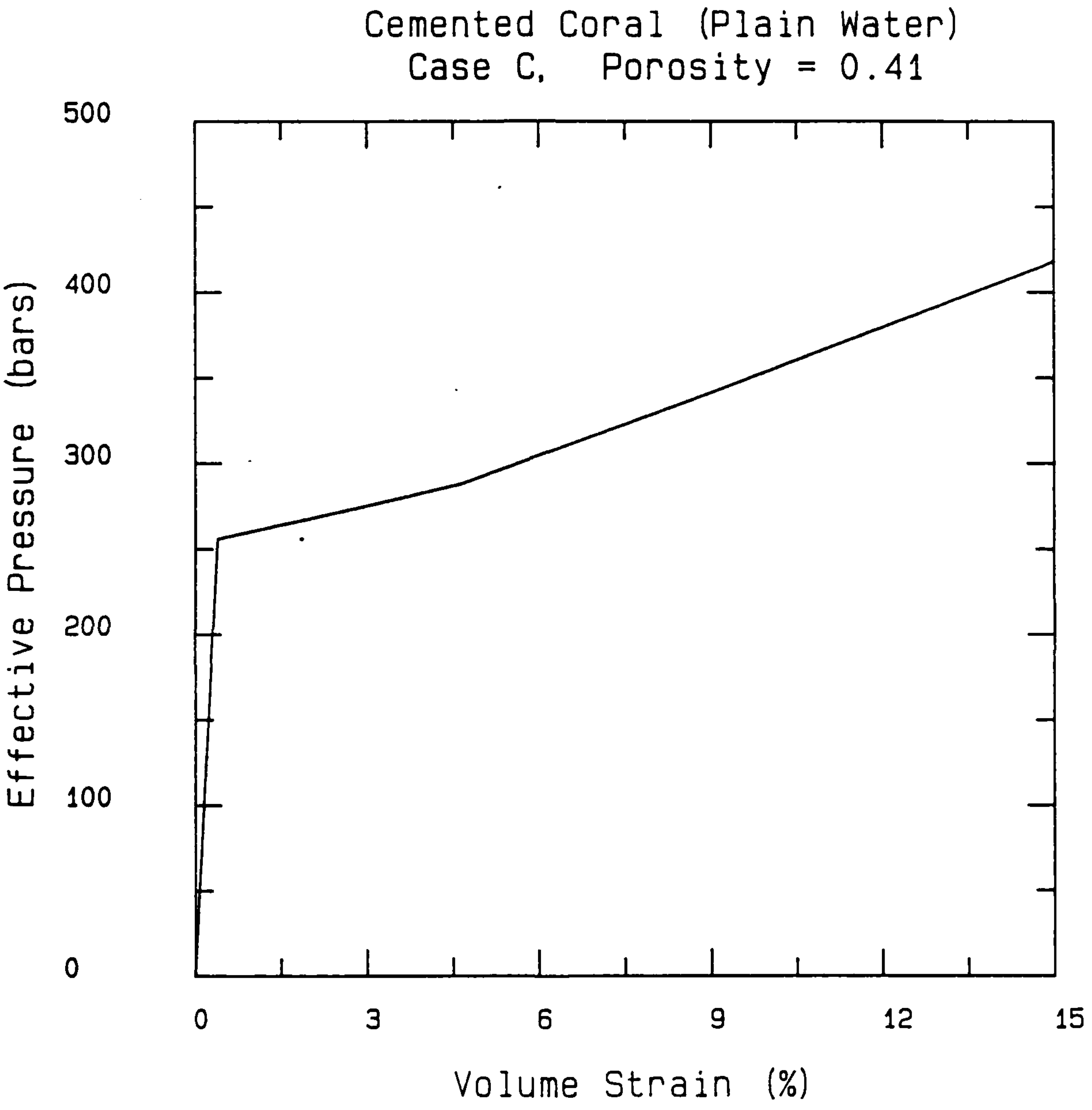
ε_i (%)	p_i (bars)	K_{fi} (bars)	p'_i (bars)	π_i (bars)	n_i
.1573E-01	.1647E+02	.1047E+06	.1024E+02	.6227E+01	.4099E+00
.3145E-01	.3293E+02	.1047E+06	.2048E+02	.1245E+02	.4099E+00
.4718E-01	.4940E+02	.1047E+06	.3072E+02	.1868E+02	.4098E+00
.6290E-01	.6586E+02	.1047E+06	.4095E+02	.2491E+02	.4098E+00
.7863E-01	.8233E+02	.1047E+06	.5119E+02	.3114E+02	.4097E+00
.9436E-01	.9880E+02	.1047E+06	.6143E+02	.3737E+02	.4097E+00
.1101E+00	.1153E+03	.1047E+06	.7167E+02	.4359E+02	.4096E+00
.1258E+00	.1317E+03	.1047E+06	.8191E+02	.4982E+02	.4096E+00
.1415E+00	.1482E+03	.1047E+06	.9215E+02	.5605E+02	.4095E+00
.1573E+00	.1647E+03	.1048E+06	.1024E+03	.6230E+02	.4095E+00
.1730E+00	.1812E+03	.1048E+06	.1126E+03	.6855E+02	.4094E+00
.1887E+00	.1977E+03	.1049E+06	.1229E+03	.7482E+02	.4094E+00
.2045E+00	.2142E+03	.1049E+06	.1331E+03	.8110E+02	.4093E+00
.2202E+00	.2307E+03	.1050E+06	.1433E+03	.8740E+02	.4093E+00
.2360E+00	.2473E+03	.1051E+06	.1536E+03	.9370E+02	.4092E+00
.2517E+00	.2638E+03	.1051E+06	.1638E+03	.1000E+03	.4092E+00
.2675E+00	.2804E+03	.1052E+06	.1741E+03	.1064E+03	.4091E+00
.2832E+00	.2970E+03	.1052E+06	.1843E+03	.1127E+03	.4091E+00
.2990E+00	.3136E+03	.1053E+06	.1945E+03	.1191E+03	.4090E+00
.3148E+00	.3302E+03	.1054E+06	.2048E+03	.1254E+03	.4089E+00
.3306E+00	.3468E+03	.1054E+06	.2150E+03	.1318E+03	.4089E+00
.3463E+00	.3635E+03	.1055E+06	.2252E+03	.1382E+03	.4088E+00
.3621E+00	.3801E+03	.1055E+06	.2355E+03	.1446E+03	.4088E+00
.3779E+00	.3968E+03	.1056E+06	.2457E+03	.1511E+03	.4087E+00
.3937E+00	.4135E+03	.1057E+06	.2560E+03	.1575E+03	.4087E+00
.5646E+00	.5006E+03	.5094E+05	.2573E+03	.2433E+03	.4078E+00
.7363E+00	.5901E+03	.5213E+05	.2586E+03	.3315E+03	.4069E+00
.9087E+00	.6821E+03	.5337E+05	.2598E+03	.4222E+03	.4061E+00
.1082E+01	.7767E+03	.5465E+05	.2611E+03	.5155E+03	.4052E+00
.1256E+01	.8741E+03	.5598E+05	.2624E+03	.6116E+03	.4043E+00
.1431E+01	.9743E+03	.5735E+05	.2637E+03	.7106E+03	.4034E+00
.1606E+01	.1078E+04	.5879E+05	.2650E+03	.8125E+03	.4026E+00
.1783E+01	.1184E+04	.6027E+05	.2663E+03	.9176E+03	.4017E+00
.1960E+01	.1294E+04	.6182E+05	.2676E+03	.1026E+04	.4008E+00
.2136E+01	.1406E+04	.6384E+05	.2689E+03	.1137E+04	.3999E+00
.2310E+01	.1520E+04	.6551E+05	.2702E+03	.1250E+04	.3990E+00
.2483E+01	.1636E+04	.6718E+05	.2715E+03	.1365E+04	.3981E+00
.2655E+01	.1754E+04	.6886E+05	.2728E+03	.1481E+04	.3972E+00
.2825E+01	.1874E+04	.7053E+05	.2741E+03	.1600E+04	.3963E+00
.2995E+01	.1998E+04	.7298E+05	.2754E+03	.1720E+04	[illegible]
.3164E+01	.2126E+04	.7564E+05	.2767E+03	.1849E+04	.3945E+00
.3332E+01	.2258E+04	.7837E+05	.2780E+03	.1980E+04	[illegible]
.3500E+01	.2395E+04	.8119E+05	.2793E+03	.2115E+04	.3926E+00
.3668E+01	.2534E+04	.8352E+05	.2806E+03	.2254E+04	.3917E+00
.3834E+01	.2677E+04	.8578E+05	.2819E+03	.2395E+04	.3908E+00
.4000E+01	.2824E+04	.8810E+05	.2832E+03	.2541E+04	.3898E+00
.4166E+01	.2973E+04	.9045E+05	.2845E+03	.2689E+04	.3889E+00
.4331E+01	.3127E+04	.9286E+05	.2858E+03	.2841E+04	.3880E+00

```
.4496E+01    .3284E+04    .9532E+05    .2871E+03    .2997E+04    .3870E+00
.4660E+01    .3445E+04    .9783E+05    .2883E+03    .3156E+04    .3861E+00
.4822E+01    .3609E+04    .1017E+06    .2903E+03    .3319E+04    .3852E+00
.4983E+01    .3779E+04    .1052E+06    .2923E+03    .3486E+04    .3842E+00
.5145E+01    .3954E+04    .1088E+06    .2942E+03    .3660E+04    .3833E+00
.5306E+01    .4136E+04    .1126E+06    .2962E+03    .3840E+04    .3824E+00
.5467E+01    .4323E+04    .1165E+06    .2981E+03    .4025E+04    .3814E+00
.5628E+01    .4517E+04    .1205E+06    .3001E+03    .4217E+04    .3805E+00
.5788E+01    .4718E+04    .1248E+06    .3021E+03    .4416E+04    .3795E+00
.5949E+01    .4925E+04    .1293E+06    .3040E+03    .4621E+04    .3786E+00
.6109E+01    .5140E+04    .1338E+06    .3060E+03    .4834E+04    .3776E+00
.6270E+01    .5362E+04    .1386E+06    .3079E+03    .5054E+04    .3766E+00
.6430E+01    .5592E+04    .1435E+06    .3099E+03    .5282E+04    .3757E+00
.6590E+01    .5830E+04    .1486E+06    .3119E+03    .5518E+04    .3747E+00
.6750E+01    .6076E+04    .1538E+06    .3138E+03    .5763E+04    .3737E+00
.6910E+01    .6331E+04    .1593E+06    .3158E+03    .6015E+04    .3728E+00
.7070E+01    .6595E+04    .1649E+06    .3177E+03    .6277E+04    .3718E+00
.7230E+01    .6868E+04    .1709E+06    .3197E+03    .6548E+04    .3708E+00
.7389E+01    .7150E+04    .1770E+06    .3217E+03    .6829E+04    .3698E+00
.7549E+01    .7443E+04    .1834E+06    .3236E+03    .7120E+04    .3688E+00
.7709E+01    .7747E+04    .1903E+06    .3256E+03    .7422E+04    .3678E+00
.7868E+01    .8063E+04    .1977E+06    .3275E+03    .7735E+04    .3668E+00
.8028E+01    .8390E+04    .2053E+06    .3295E+03    .8061E+04    .3658E+00
.8188E+01    .8731E+04    .2134E+06    .3315E+03    .8400E+04    .3648E+00
.8347E+01    .9086E+04    .2226E+06    .3334E+03    .8753E+04    .3638E+00
.8507E+01    .9457E+04    .2322E+06    .3354E+03    .9122E+04    .3628E+00
.8667E+01    .9845E+04    .2426E+06    .3373E+03    .9508E+04    .3618E+00
.8915E+01    .1048E+05    .2543E+06    .3405E+03    .1014E+05    .3602E+00
.9164E+01    .1115E+05    .2712E+06    .3436E+03    .1081E+05    .3587E+00
.9413E+01    .1185E+05    .2812E+06    .3467E+03    .1150E+05    .3571E+00
.9660E+01    .1257E+05    .2915E+06    .3498E+03    .1222E+05    .3554E+00
.9907E+01    .1332E+05    .3021E+06    .3530E+03    .1297E+05    .3538E+00
.1015E+02    .1409E+05    .3130E+06    .3561E+03    .1373E+05    .3522E+00
.1040E+02    .1489E+05    .3244E+06    .3592E+03    .1453E+05    .3506E+00
.1065E+02    .1571E+05    .3361E+06    .3624E+03    .1535E+05    .3489E+00
.1089E+02    .1657E+05    .3500E+06    .3655E+03    .1621E+05    .3473E+00
.1114E+02    .1747E+05    .3672E+06    .3686E+03    .1710E+05    .3456E+00
.1138E+02    .1842E+05    .3852E+06    .3717E+03    .1804E+05    .3440E+00
.1163E+02    .1941E+05    .4041E+06    .3749E+03    .1903E+05    .3423E+00
.1187E+02    .2044E+05    .4239E+06    .3780E+03    .2007E+05    .3406E+00
.1212E+02    .2153E+05    .4449E+06    .3811E+03    .2115E+05    .3389E+00
.1236E+02    .2269E+05    .4715E+06    .3843E+03    .2230E+05    .3372E+00
.1261E+02    .2391E+05    .4996E+06    .3874E+03    .2352E+05    .3355E+00
.1285E+02    .2521E+05    .5294E+06    .3905E+03    .2482E+05    .3337E+00
.1310E+02    .2659E+05    .5611E+06    .3936E+03    .2619E+05    .3320E+00
.1334E+02    .2805E+05    .5950E+06    .3968E+03    .2765E+05    .3301E+00
.1359E+02    .2960E+05    .6310E+06    .3999E+03    .2920E+05    .3285E+00
.1383E+02    .3124E+05    .6692E+06    .4030E+03    .3084E+05    .3267E+00
.1408E+02    .3297E+05    .7042E+06    .4062E+03    .3257E+05    .3250E+00
.1432E+02    .3478E+05    .7353E+06    .4093E+03    .3437E+05    .3232E+00
.1457E+02    .3666E+05    .7678E+06    .4124E+03    .3625E+05    .3214E+00
.1481E+02    .3862E+05    .8016E+06    .4155E+03    .3821E+05    .3196E+00
```

```
.1502E+02   .4038E+05   .8372E+06   .4188E+03   .3996E+05   .3180E+00
.1523E+02   .4220E+05   .8687E+06   .4220E+03   .4178E+05   .3165E+00
.1544E+02   .4412E+05   .9130E+06   .4252E+03   .4369E+05   .3149E+00
.1565E+02   .4613E+05   .9598E+06   .4284E+03   .4571E+05   .3133E+00
.1586E+02   .4825E+05   .1009E+07   .4317E+03   .4782E+05   .3117E+00
.1607E+02   .5048E+05   .1061E+07   .4349E+03   .5005E+05   .3101E+00
.1628E+02   .5283E+05   .1115E+07   .4381E+03   .5239E+05   .3085E+00
.1649E+02   .5529E+05   .1171E+07   .4413E+03   .5485E+05   .3069E+00
.1670E+02   .5788E+05   .1230E+07   .4445E+03   .5743E+05   .3053E+00
.1692E+02   .6060E+05   .1292E+07   .4478E+03   .6015E+05   .3037E+00
.1713E+02   .6346E+05   .1357E+07   .4510E+03   .6301E+05   .3020E+00
.1734E+02   .6645E+05   .1423E+07   .4542E+03   .6600E+05   .3004E+00
.1755E+02   .6960E+05   .1492E+07   .4574E+03   .6914E+05   .2987E+00
.1776E+02   .7289E+05   .1565E+07   .4607E+03   .7243E+05   .2971E+00
.1797E+02   .7635E+05   .1642E+07   .4639E+03   .7589E+05   .2954E+00
.1818E+02   .7999E+05   .1724E+07   .4671E+03   .7952E+05   .2937E+00
.1839E+02   .8380E+05   .1810E+07   .4703E+03   .8333E+05   .2920E+00
.1860E+02   .8780E+05   .1896E+07   .4736E+03   .8732E+05   .2903E+00
.1881E+02   .9198E+05   .1986E+07   .4768E+03   .9151E+05   .2886E+00
.1902E+02   .9636E+05   .2076E+07   .4800E+03   .9588E+05   .2869E+00
.1923E+02   .1009E+06   .2164E+07   .4832E+03   .1004E+06   .2852E+00
.1944E+02   .1057E+06   .2251E+07   .4865E+03   .1052E+06   .2834E+00
.1965E+02   .1105E+06   .2302E+07   .4897E+03   .1100E+06   .2817E+00
.1986E+02   .1154E+06   .2353E+07   .4929E+03   .1143E+06   .2800E+00
.2007E+02   .1205E+06   .2406E+07   .4961E+03   .1200E+06   .2782E+00
```



Cemented Coral (Plain Water)
Case C, Porosity = 0.41


Cemented Coral (Plain Water)
Case C, Porosity = 0.41
Effective Pressure (bars)
0
100
200
300
400
500
0
3
6
9
12
15
Volume Strain (%)

APPENDIX D

UNDRAINED BEHAVIOR OF FULLY COUPLED MODEL IN UNIAXIAL COMPRESSION

The computer program KOCP, listed in subsection D.1, uses input from TAPES 5, 7 and 8. TAPE7 contains the water modulus data described in Table 5.1. TAPE5 contains the initial porosity and a table of effective axial skeleton stress as a function of effective vertical strain. TAPE8 contains a table of k_o vs effective axial strain pairs. The compressibility of the solid grains from Section 5.2.2 is built into subroutine BULKG.

DESCRIPTION OF INPUT FILES

TAPE7 - See Appendix A (TAPE6)

TAPE5

Card 1

POR, NPOINT, NDIV, EMAX (free format)

POR = Initial porosity
NPOINT = Number of effective axial strain/skeleton constrained modulus pairs
NDIV = Total number of steps
EMAX = Maximum total strain at which calculation terminates

Card 2

$$\left.\begin{array}{l} \epsilon'_1,\ \sigma'_{a1} \\ \epsilon'_2,\ \sigma'_{a2} \\ -\quad - \\ -\quad - \\ \epsilon'_n,\ \sigma'_{an} \end{array}\right\}$$

NPOINT cards with free format for each card.
Note that σ'_{ai} is in terms of bars.

TAPE8

Card 1

NKO (free format)

NKO = Number of effective axial strain/k_o pairs

Card 2

$$\left.\begin{array}{l} \epsilon'_1,\ k_{o1} \\ \epsilon'_i,\ k_{o2} \\ -\quad - \\ -\quad - \\ \epsilon'_n,\ k_{on} \end{array}\right\} \quad \text{NKO cards with free format for each card}$$

The output is contained in TAPE3 and 6. TAPEs 3 and 6 list total axial stress (σ_{ai}), total radial stress (σ_{ri}), undrained constrained modulus (M_{fi}), effective axial stress (σ'_{ai}), effective radial stress (σ'_{ri}), pore water pressure (π_i), and current porosity (n_i) as functions of total axial strain (ϵ_{ai}).

D.1 List of Program KOCP

```
$FILES(0.3)
      PROGRAM KOCP
C
C REVISION: October 10, 1985
C REVISION: October 11, 1985
C REVISION: October 23, 1985
C REVISION: October 30, 1985
C REVISION: March 10, 1986
C REVISION: March 11, 1986 (Now writes out true strain)
C REVISION: July 28, 1986 (deleted the above and changed BMF
C                equation at line 123)
C
      COMMON/WATER/PP(100),BM(100)
      COMMON/SKTON/RR(100),CM(100)
      COMMON/KODAT/EKO(100),RKO(100)
      INTEGER NAM1(6),NAM2(6),NAM3(6),NAM4(6),NAM5(6)
C
C Read bulk modulus of water as a function of pressure.
C
      WRITE(1,*)'     '
      WRITE(1,*)'The PRESSURE-BULK MODULUS file has # of data pairs'
      WRITE(1,*)'    at the top with pairs in bars '
      WRITE(1,*)'The STRAIN-STRESS file has INITIAL POROSITY, # of    '
      WRITE(1,*)'    data pairs, # of divi., and max strain at top'
      WRITE(1,*)'      '
      WRITE(1,*)'The Ko file has # of data points at the top '
      WRITE(1,*)'    then pairs of EFFECTIVE VERTICAL STRAIN vs '
      WRITE(1,*)'    VALUE of Ko (must have at least two pairs)'
      WRITE(1,*)'          '
      WRITE(1,*)'**NOTE: STRAIN-STRESS data must begin with zero  '
      WRITE(1,*)'**NOTE: The first EFFECTIVE VERTICAL STRAIN '
      WRITE(1,*)'        must be zero and the last .GE. EMAX'
      WRITE(1,*)'        '
      WRITE(1)'Type INPUT file with PRESSURE-BULK MODULUS pairs  '
      READ(1,10) NAM1
10    FORMAT(6A2)
      WRITE(1)'Type INPUT file with STRESS-STRAIN pairs          '
      READ(1,10) NAM2
      WRITE(1)'Type INPUT file with EFFECTIVE VERT. STRAIN vs Ko '
      READ(1,10) NAM4
      OPEN(7,FILE=NAM1)
      WRITE(1)'Type OUTPUT file (STRN.BMF,PF,POR)                '
      READ(1,10) NAM3
      WRITE(1)'Type OUTPUT file (STRN.SE,S.SEH.SH)               '
      READ(1,10) NAM5
      WRITE(1,*)'                  '
      WRITE(1)'Type case A,B or C                                '
      READ(1,30) IANS
30    FORMAT(A1)
```

```
C
      OPEN(6,FILE=NAM3)
      OPEN(3,FILE=NAM5)
      READ(7,*) NWATER
      DO 100 I=1,NWATER
100   READ(7,*) PP(I),BM(I)
      CLOSE(7)
C
C Read variable Ko as a function of effective vertical strain
C
      OPEN(8,FILE=NAM4)
C
      READ(8,*) NKO
      DO 50 I=1,NKO
50    READ(8,*) EKO(I),RKO(I)
      CLOSE(8)
C
C Read initial porosity and # of data pairs in STRESS-STRAIN curve
C
      OPEN(5,FILE=NAM2)
      READ(5,*) POR,NPOINT,NDIV,EMAX
C
      DO 120 I=1,NPOINT
120   READ(5,*)RR(I),CM(I)
C
      CLOSE (5)
C
C Set initial conditions
C
      VO = 1.0
      PORI = POR
      STRN = 0.0
      EE = 0.0
      S = 0.0
      SE = 0.0
      PE = 0.0
      PF = 0.0
      NFIRST = 0
      KK = 1
C
      OPEN(6,FILE=NAM3)
C
C Steo-by-steo calculations
C
 150  IF(NFIRST.EQ.0) GOTO 200
      K = K + 1
      IF(K.EQ.NDIV) GOTO 200
      GOTO 300
200   CONTINUE
      K = 0
      NFIRST = 1
      KK = KK + 1
      DDTRN = (RR(KK)-RR(KK-1))/NDIV
300   CONTINUE
```

```
C
C Calculate skeleton constrained modulus
C
            CALL CONMS(EE,BMS)
C
C Calculate skeleton bulk modulus
C
            CALL BULKS(BMS,BKS,EE)
C
C Calculate grain bulk modulus
C
            CALL BULKG(PF,PE,POR,BKG,IANS)
C
C Calculate water bulk modulus
C
            CALL BULKW(PF,BKW)
C
C Calculate undrained constrained modulus
C
            AA = (1.-PORI)/(1.-POR)
            BMF = BMS+(BKG-AA*BKS)/(1.+PORI*BKG*(BKG-BKW)/
     #            (BKW*(BKG-AA*BKS)))
C
C Calculate total strain increment
C
             DSTRN = DDTRN*BMS*(BKG-BKS)/(BMS*BKG-BMF*BKS)
C
C Calculate oressure increments
C
            DS = BMF*DSTRN
            DPF = DSTRN*(BMF-BMS)/(1.-BKS/BKG)
            DSE = DS - DPF
            DPE = BKS*(DSTRN-DPF/BKG)
C
C Uodate oressures
C
            PF = PF+DPF
            PE = PE+DPE
            S = S + DS
            SE = SE + DSE
            SEH = 1.5 * PE- 0.5 * SE
            SH = SEH+PF
C
C Uodate strain, volume of water and oorosity
C
            EE = EE + DDTRN
            STRNO = STRN
            PORO = POR
            STRN = STRN +DSTRN
            DVW = PORO*(1.-STRNO)-PORI*DPF/BKW
            POR = DVW/(1.-STRN)
C
```

```
C Write:
C           Total vertical stress (S)
C           Total horizontal stress (SH)
C           Undrained constrained modulus (BMF)
C           Effective vertical stress (SE)
C           Effective horizontal stress (SEH)
C           Pore water pressure (PF)
C           and porosity (POR)
C           as a function of strain (STRN)
C
            WRITE(6,2001) (STRN *100),BMF,PF,POR
            WRITE(3,*) (STRN *100),SE,S,SEH,SH
2001        FORMAT(4E12.4)
            IF(STRN.GT.EMAX) GOTO 500
            GOTO 150
 500  CONTINUE
      CLOSE(6)
      CLOSE(3)
      STOP
      END
C
C************************************************
C
      SUBROUTINE BULKG(PF,PE,POR,BKG,IANS)
C
C Assume constant
C
      IF(IANS.NE.1HA) THEN
      P = PF+PE/(1.-POR)
      IF(P.LE.1363.) BKG = 344827.
      IF(P.GT.1363.) BKG = 253. * P
      ENDIF
      IF(IANS.EQ.1HB) THEN
      IF(BKG.GT.688000.) BKG = 344827.
      ENDIF
      IF(IANS.EQ.1HA)  BKG = 413793.  !In bars
      RETURN
      END
C
C------------------------------------------------------
C
      SUBROUTINE BULKW(PF,BKW)
C
      COMMON/WATER/PP(100),BM(100)
C
      I = 1
100   CONTINUE
      IF(PF.GE.PP(I).AND.PF.LE.PP(I+1)) GOTO 200
      I = I+1
      GOTO 100
200   CONTINUE
      BKW = BM(I)+(BM(I+1)-BM(I))*(PF-PP(I))/(PP(I+1)-PP(I))
      RETURN
      END
C
```

```
C------------------------------------------------------------
C
      SUBROUTINE BULKS(BMS,BKS,EE)
C
C Find Ko
C
      CALL CALKO(EE,XKO)
      A = (2.0*XKO+1.0)/3.
      BKS = A*BMS
      RETURN
      END
C
C----------------------------------------------------------
C
      SUBROUTINE CONMS(EE,BMS)
      COMMON/SKTON/RR(100),CM(100)
      I = 1
100   CONTINUE
      IF(EE.GE.RR(I).AND.EE.LE.RR(I+1)) GOTO 200
      I = I +1
      GOTO 100
200   CONTINUE
      BMS = (CM(I+1)-CM(I))/(RR(I+1)-RR(I))
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE CALKO(EE,XKO)
      COMMON/KODAT/EKO(100),RKO(100)
      I = 1
100   CONTINUE
      IF(EE.GE.EKO(I).AND.EE.LE.EKO(I+1)) GOTO 200
      I = I +1
      GOTO 100
200   CONTINUE
      XKO = RKO(I)
      RETURN
      END
C
```

D.2 SAMPLE PROBLEMS

Two sample problems were prepared to illustrate the use of KOCP in uncemented and cemented porous media.

D.2.1 Beach Sand

TAPE5 contains the input for the skeleton in tabular form as effective axial strain-effective axial stress pairs.

TAPE5

```
0.4           15           50           .200

0.0            0.0
3.4E-2       115.48
6.2E-2       210.58
9.1E-2       309.08
 .115        390.6
 .161        651.42
 .187        851.76
 .216       1229.8
 .235       1547.3
 .249       1893.8
 .264       2346.1
 .277       2928.2
 .288       3588.5
 .298       4413.8
 .301       4779.2
```

TAPE8 lists the effective axial strain-k_o pairs.

TAPE8

```
4

0.0          .1
 .00348      .75
 .157        .47
 .30         .47
```

TAPEs 3 and 6 contain the output data: σ_{ai} (bars), σ_{ri} (bars), M_{fi} (bars), σ'_{ai} (bars), σ'_{ri} (bars), π_i (bars), n_i and ϵ_{ai}.

List of Output File Tape 3

ε_{ai}	M_{fi} (bars)	τ_i (bars)	n_i
.1504E+00	.5265E+05	.7457E+02	.3992E+00
.3009E+00	.5299E+05	.1497E+03	.3985E+00
.4518E+00	.5403E+05	.2266E+03	.3977E+00
.6030E+00	.5511E+05	.3053E+03	.3969E+00
.7546E+00	.5622E+05	.3859E+03	.3962E+00
.9066E+00	.5737E+05	.4685E+03	.3954E+00
.1059E+01	.5855E+05	.5531E+03	.3946E+00
.1212E+01	.5977E+05	.6398E+03	.3938E+00
.1365E+01	.6103E+05	.7286E+03	.3931E+00
.1518E+01	.6233E+05	.8198E+03	.3923E+00
.1673E+01	.6368E+05	.9133E+03	.3915E+00
.1827E+01	.6507E+05	.1009E+04	.3907E+00
.1982E+01	.6646E+05	.1108E+04	.3899E+00
.2138E+01	.6751E+05	.1208E+04	.3892E+00
.2293E+01	.6857E+05	.1310E+04	.3884E+00
.2449E+01	.6976E+05	.1414E+04	.3876E+00
.2604E+01	.7149E+05	.1520E+04	.3868E+00
.2758E+01	.7335E+05	.1628E+04	.3860E+00
.2911E+01	.7587E+05	.1740E+04	.3852E+00
.3063E+01	.7843E+05	.1855E+04	.3844E+00
.3216E+01	.8106E+05	.1974E+04	.3836E+00
.3367E+01	.8376E+05	.2096E+04	.3828E+00
.3518E+01	.8604E+05	.2221E+04	.3819E+00
.3669E+01	.8823E+05	.2349E+04	.3811E+00
.3819E+01	.9045E+05	.2481E+04	.3803E+00
.3942E+01	.9270E+05	.2591E+04	.3796E+00
.4065E+01	.9458E+05	.2703E+04	.3789E+00
.4187E+01	.9649E+05	.2818E+04	.3782E+00
.4310E+01	.9843E+05	.2935E+04	.3775E+00
.4432E+01	.1004E+06	.3054E+04	.3768E+00
.4554E+01	.1027E+06	.3175E+04	.3761E+00
.4676E+01	.1054E+06	.3300E+04	.3754E+00
.4798E+01	.1082E+06	.3428E+04	.3747E+00
.4920E+01	.1111E+06	.3559E+04	.3740E+00
.5041E+01	.1140E+06	.3694E+04	.3733E+00
.5163E+01	.1170E+06	.3832E+04	.3726E+00
.5284E+01	.1201E+06	.3974E+04	.3719E+00
.5405E+01	.1233E+06	.4120E+04	.3712E+00
.5526E+01	.1266E+06	.4270E+04	.3705E+00
.5648E+01	.1300E+06	.4423E+04	.3697E+00
.5769E+01	.1336E+06	.4581E+04	.3690E+00
.5890E+01	.1372E+06	.4744E+04	.3683E+00
.6011E+01	.1409E+06	.4910E+04	.3675E+00
.6131E+01	.1447E+06	.5081E+04	.3668E+00
.6252E+01	.1486E+06	.5257E+04	.3661E+00
.6373E+01	.1526E+06	.5437E+04	.3654E+00
.6494E+01	.1567E+06	.5623E+04	.3646E+00

```
.6614E+01    .1610E+06    .5813E+04    .3639E+00
.6735E+01    .1653E+06    .6009E+04    .3632E+00
.6855E+01    .1698E+06    .6209E+04    .3624E+00
.6980E+01    .1745E+06    .6423E+04    .3617E+00
.7105E+01    .1795E+06    .6643E+04    .3609E+00
.7229E+01    .1846E+06    .6869E+04    .3601E+00
.7354E+01    .1899E+06    .7102E+04    .3593E+00
.7479E+01    .1955E+06    .7342E+04    .3586E+00
.7603E+01    .2015E+06    .7589E+04    .3578E+00
.7728E+01    .2077E+06    .7844E+04    .3570E+00
.7852E+01    .2140E+06    .8106E+04    .3562E+00
.7977E+01    .2209E+06    .8377E+04    .3554E+00
.8102E+01    .2284E+06    .8658E+04    .3546E+00
.8226E+01    .2362E+06    .8948E+04    .3538E+00
.8351E+01    .2442E+06    .9249E+04    .3531E+00
.8475E+01    .2534E+06    .9561E+04    .3523E+00
.8600E+01    .2630E+06    .9885E+04    .3515E+00
.8725E+01    .2730E+06    .1022E+05    .3507E+00
.8850E+01    .2799E+06    .1057E+05    .3499E+00
.8974E+01    .2852E+06    .1092E+05    .3491E+00
.9099E+01    .2906E+06    .1128E+05    .3483E+00
.9223E+01    .2960E+06    .1164E+05    .3474E+00
.9347E+01    .3015E+06    .1201E+05    .3466E+00
.9471E+01    .3071E+06    .1239E+05    .3458E+00
.9595E+01    .3129E+06    .1277E+05    .3450E+00
.9719E+01    .3187E+06    .1316E+05    .3442E+00
.9843E+01    .3246E+06    .1356E+05    .3434E+00
.9967E+01    .3306E+06    .1396E+05    .3425E+00
.1007E+02    .3367E+06    .1431E+05    .3418E+00
.1017E+02    .3419E+06    .1465E+05    .3412E+00
.1027E+02    .3471E+06    .1500E+05    .3405E+00
.1038E+02    .3524E+06    .1536E+05    .3398E+00
.1048E+02    .3597E+06    .1572E+05    .3391E+00
.1058E+02    .3672E+06    .1610E+05    .3384E+00
.1068E+02    .3748E+06    .1648E+05    .3377E+00
.1078E+02    .3826E+06    .1686E+05    .3370E+00
.1089E+02    .3906E+06    .1726E+05    .3363E+00
.1099E+02    .3987E+06    .1766E+05    .3356E+00
.1109E+02    .4070E+06    .1807E+05    .3349E+00
.1119E+02    .4155E+06    .1849E+05    .3342E+00
.1129E+02    .4241E+06    .1892E+05    .3335E+00
.1140E+02    .4329E+06    .1936E+05    .3328E+00
.1150E+02    .4419E+06    .1981E+05    .3321E+00
.1160E+02    .4511E+06    .2027E+05    .3314E+00
.1170E+02    .4617E+06    .2073E+05    .3307E+00
.1180E+02    .4734E+06    .2121E+05    .3300E+00
.1191E+02    .4853E+06    .2171E+05    .3293E+00
.1201E+02    .4976E+06    .2221E+05    .3285E+00
.1211E+02    .5102E+06    .2273E+05    .3279E+00
.1221E+02    .5231E+06    .2326E+05    .3271E+00
.1231E+02    .5364E+06    .2380E+05    .3264E+00
.1242E+02    .5500E+06    .2436E+05    .3256E+00
.1252E+02    .5639E+06    .2493E+05    .3249E+00
```

```
.1271E+02    .5782E+06    .2606E+05    .3235E+00
.1291E+02    .6087E+06    .2724E+05    .3221E+00
.1311E+02    .6386E+06    .2848E+05    .3207E+00
.1330E+02    .6699E+06    .2978E+05    .3193E+00
.1350E+02    .7028E+06    .3115E+05    .3179E+00
.1369E+02    .7295E+06    .3257E+05    .3165E+00
.1389E+02    .7557E+06    .3403E+05    .3150E+00
.1408E+02    .7828E+06    .3556E+05    .3136E+00
.1428E+02    .8108E+06    .3713E+05    .3122E+00
.1448E+02    .8399E+06    .3876E+05    .3107E+00
.1467E+02    .8699E+06    .4045E+05    .3093E+00
.1487E+02    .9040E+06    .4221E+05    .3078E+00
.1506E+02    .9479E+06    .4405E+05    .3063E+00
.1526E+02    .9940E+06    .4598E+05    .3048E+00
.1545E+02    .1042E+07    .4801E+05    .3034E+00
.1565E+02    .1093E+07    .5013E+05    .3019E+00
.1584E+02    .1146E+07    .5236E+05    .3004E+00
.1604E+02    .1201E+07    .5470E+05    .2989E+00
.1624E+02    .1258E+07    .5716E+05    .2974E+00
.1643E+02    .1318E+07    .5973E+05    .2958E+00
.1663E+02    .1381E+07    .6242E+05    .2943E+00
.1682E+02    .1445E+07    .6524E+05    .2928E+00
.1702E+02    .1512E+07    .6819E+05    .2912E+00
.1721E+02    .1581E+07    .7127E+05    .2897E+00
.1741E+02    .1655E+07    .7451E+05    .2881E+00
.1752E+02    .1733E+07    .7642E+05    .2873E+00
.1763E+02    .1781E+07    .7838E+05    .2864E+00
.1774E+02    .1829E+07    .8040E+05    .2855E+00
.1785E+02    .1877E+07    .8247E+05    .2846E+00
.1796E+02    .1925E+07    .8459E+05    .2837E+00
.1807E+02    .1974E+07    .8677E+05    .2828E+00
.1818E+02    .2024E+07    .8900E+05    .2819E+00
.1830E+02    .2075E+07    .9129E+05    .2810E+00
.1841E+02    .2125E+07    .9364E+05    .2801E+00
.1852E+02    .2173E+07    .9603E+05    .2792E+00
.1863E+02    .2222E+07    .9849E+05    .2783E+00
.1874E+02    .2272E+07    .1010E+06    .2774E+00
.1885E+02    .2314E+07    .1035E+06    .2765E+00
.1896E+02    .2342E+07    .1061E+06    .2756E+00
.1907E+02    .2370E+07    .1087E+06    .2746E+00
.1918E+02    .2399E+07    .1114E+06    .2737E+00
.1929E+02    .2427E+07    .1140E+06    .2728E+00
.1940E+02    .2456E+07    .1167E+06    .2719E+00
.1951E+02    .2486E+07    .1195E+06    .2710E+00
.1962E+02    .2516E+07    .1222E+06    .2700E+00
.1973E+02    .2546E+07    .1250E+06    .2691E+00
.1984E+02    .2576E+07    .1278E+06    .2682E+00
.1995E+02    .2607E+07    .1307E+06    .2672E+00
.2006E+02    .2638E+07    .1336E+06    .2663E+00
```

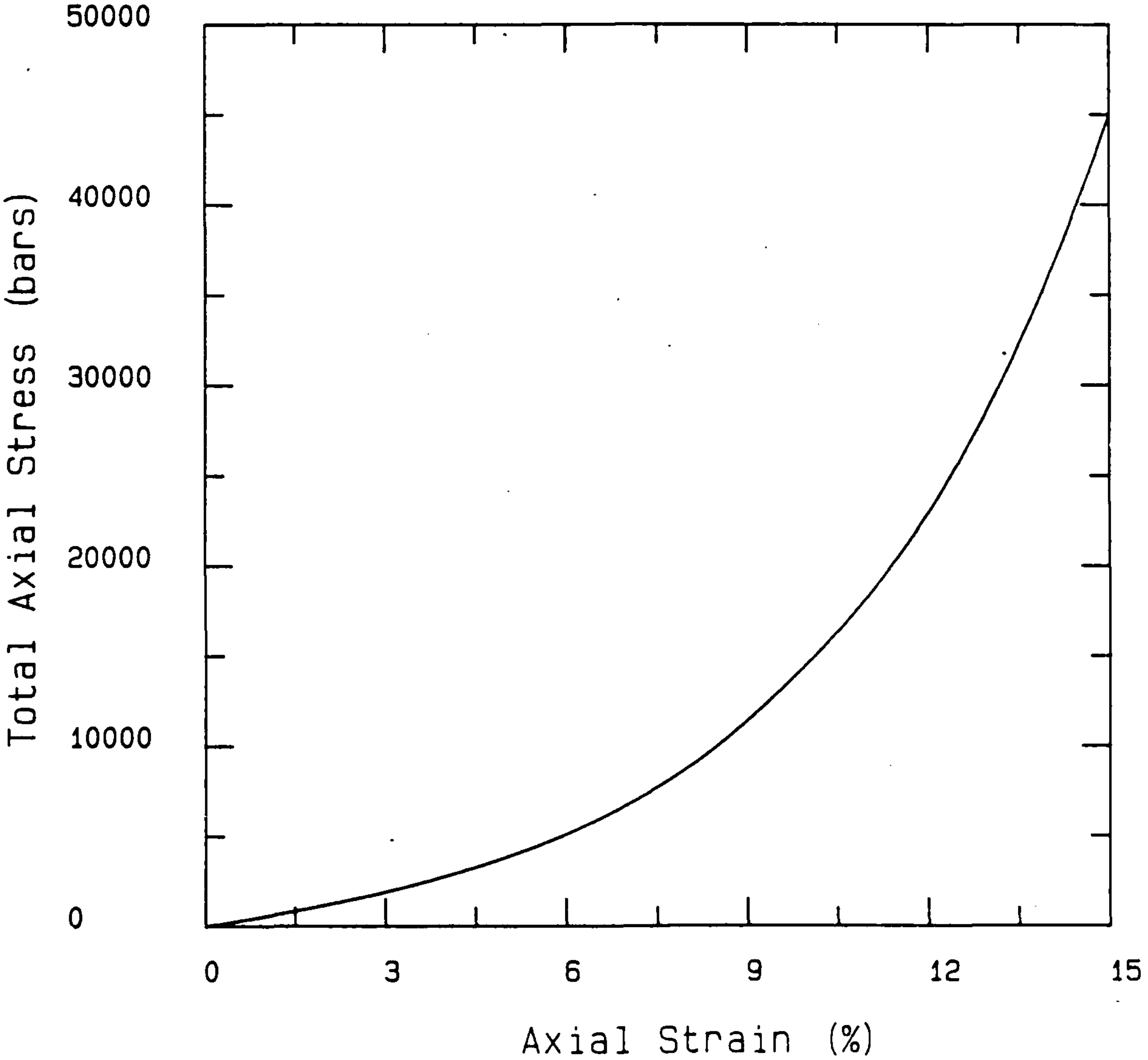
List of Output File Tape 6

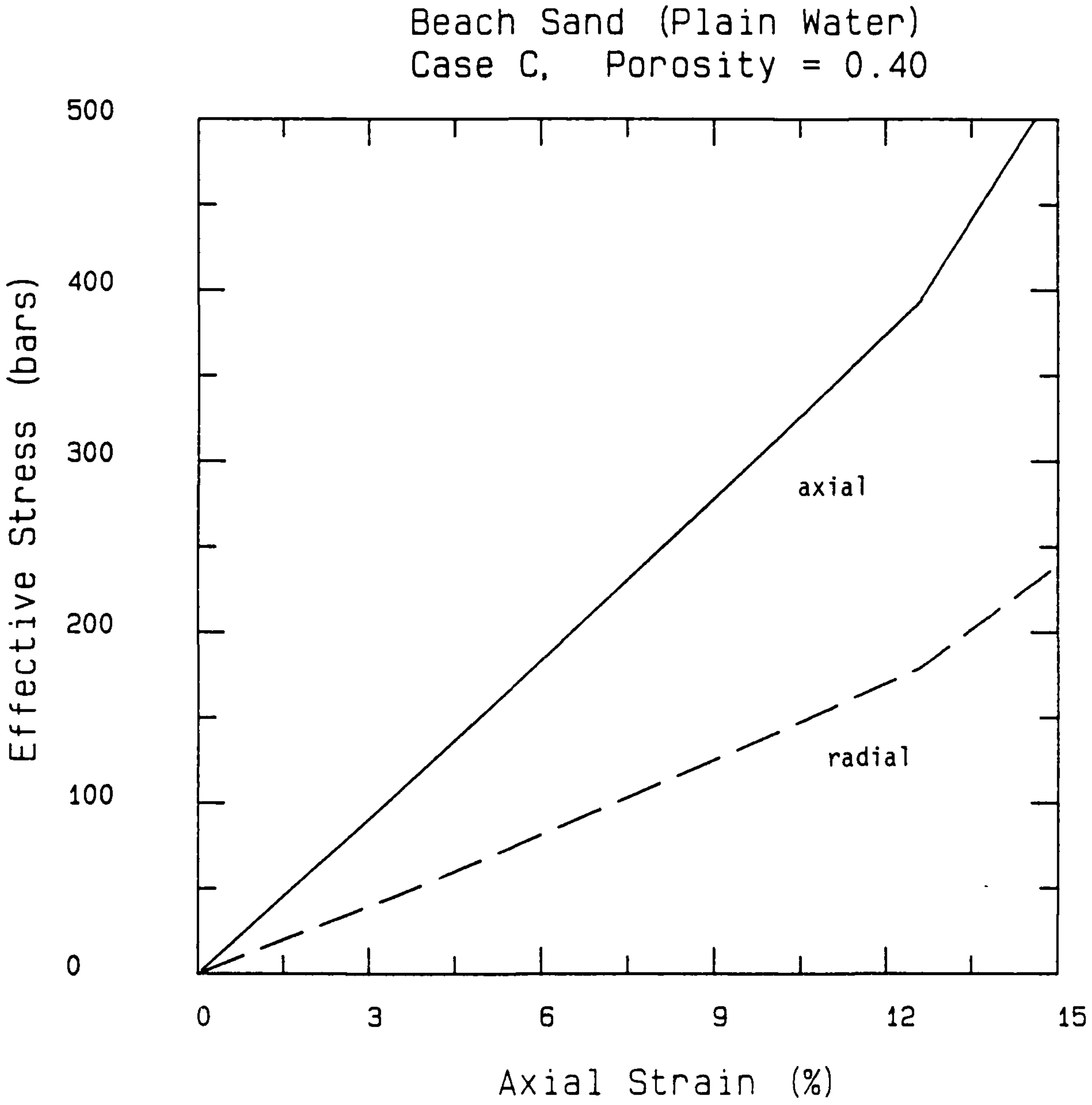
ε_{ai}	σ'_{ai} (bars)	σ_{ai} (bars)	σ'_{ri} (bars)	σ_{ri} (bars)
.1504E+00	.4619E+01	.7919E+02	.2065E+01	.7664E+02
.3009E+00	.9238E+01	.1590E+03	.4128E+01	.1538E+03
.4518E+00	.1386E+02	.2405E+03	.6185E+01	.2328E+03
.6030E+00	.1848E+02	.3238E+03	.8236E+01	.3136E+03
.7546E+00	.2310E+02	.4090E+03	.1028E+02	.3962E+03
.9066E+00	.2772E+02	.4962E+03	.1232E+02	.4808E+03
.1059E+01	.3233E+02	.5854E+03	.1435E+02	.5674E+03
.1212E+01	.3695E+02	.6767E+03	.1638E+02	.6561E+03
.1365E+01	.4157E+02	.7702E+03	.1839E+02	.7470E+03
.1518E+01	.4619E+02	.8660E+03	.2040E+02	.8402E+03
.1673E+01	.5081E+02	.9641E+03	.2241E+02	.9357E+03
.1827E+01	.5543E+02	.1065E+04	.2440E+02	.1034E+04
.1982E+01	.6005E+02	.1168E+04	.2639E+02	.1134E+04
.2138E+01	.6467E+02	.1273E+04	.2837E+02	.1236E+04
.2293E+01	.6929E+02	.1379E+04	.3034E+02	.1340E+04
.2449E+01	.7391E+02	.1488E+04	.3231E+02	.1447E+04
.2604E+01	.7853E+02	.1599E+04	.3431E+02	.1555E+04
.2758E+01	.8315E+02	.1712E+04	.3631E+02	.1665E+04
.2911E+01	.8776E+02	.1828E+04	.3833E+02	.1778E+04
.3063E+01	.9238E+02	.1947E+04	.4036E+02	.1895E+04
.3216E+01	.9700E+02	.2071E+04	.4240E+02	.2016E+04
.3367E+01	.1016E+03	.2198E+04	.4444E+02	.2140E+04
.3518E+01	.1062E+03	.2328E+04	.4650E+02	.2268E+04
.3669E+01	.1109E+03	.2460E+04	.4856E+02	.2398E+04
.3819E+01	.1155E+03	.2596E+04	.5063E+02	.2531E+04
.3942E+01	.1193E+03	.2710E+04	.5234E+02	.2643E+04
.4065E+01	.1231E+03	.2827E+04	.5406E+02	.2757E+04
.4187E+01	.1269E+03	.2945E+04	.5578E+02	.2874E+04
.4310E+01	.1307E+03	.3065E+04	.5750E+02	.2992E+04
.4432E+01	.1345E+03	.3188E+04	.5923E+02	.3113E+04
.4554E+01	.1383E+03	.3313E+04	.6096E+02	.3236E+04
.4676E+01	.1421E+03	.3442E+04	.6270E+02	.3363E+04
.4798E+01	.1459E+03	.3574E+04	.6443E+02	.3492E+04
.4920E+01	.1497E+03	.3709E+04	.6617E+02	.3625E+04
.5041E+01	.1535E+03	.3848E+04	.6791E+02	.3762E+04
.5163E+01	.1573E+03	.3990E+04	.6965E+02	.3902E+04
.5284E+01	.1611E+03	.4136E+04	.7139E+02	.4043E+04
.5405E+01	.1649E+03	.4285E+04	.7314E+02	.4193E+04
.5526E+01	.1687E+03	.4437E+04	.7488E+02	.4347E+04
.5648E+01	.1725E+03	.4596E+04	.7663E+02	.4502E+04
.5769E+01	.1763E+03	.4758E+04	.7838E+02	.4662E+04
.5890E+01	.1801E+03	.4924E+04	.8013E+02	.4824E+04
.6011E+01	.1840E+03	.5094E+04	.8188E+02	.4992E+04
.6131E+01	.1878E+03	.5269E+04	.8363E+02	.5165E+04
.6252E+01	.1916E+03	.5448E+04	.8538E+02	.5342E+04
.6373E+01	.1954E+03	.5633E+04	.8714E+02	.5524E+04
.6494E+01	.1992E+03	.5822E+04	.8889E+02	.5712E+04
.6614E+01	.2030E+03	.6016E+04	.9065E+02	.5904E+04

```
.6735E+01   .2068E+03   .6215E+04   .9241E+02   .6101E+04
.6855E+01   .2106E+03   .6420E+04   .9416E+02   .6304E+04
.6980E+01   .2145E+03   .6636E+04   .9598E+02   .6517E+04
.7105E+01   .2185E+03   .6861E+04   .9781E+02   .6741E+04
.7229E+01   .2224E+03   .7092E+04   .9963E+02   .6967E+04
.7354E+01   .2263E+03   .7328E+04   .1015E+03   .7203E+04
.7479E+01   .2303E+03   .7572E+04   .1033E+03   .7445E+04
.7603E+01   .2342E+03   .7823E+04   .1051E+03   .7694E+04
.7728E+01   .2382E+03   .8082E+04   .1069E+03   .7950E+04
.7852E+01   .2421E+03   .8348E+04   .1087E+03   .8215E+04
.7977E+01   .2460E+03   .8623E+04   .1106E+03   .8488E+04
.8102E+01   .2500E+03   .8908E+04   .1124E+03   .8770E+04
.8226E+01   .2539E+03   .9202E+04   .1142E+03   .9063E+04
.8351E+01   .2579E+03   .9507E+04   .1160E+03   .9365E+04
.8475E+01   .2618E+03   .9823E+04   .1179E+03   .9679E+04
.8600E+01   .2657E+03   .1015E+05   .1197E+03   .1000E+05
.8725E+01   .2697E+03   .1049E+05   .1215E+03   .1034E+05
.8850E+01   .2736E+03   .1084E+05   .1233E+03   .1069E+05
.8974E+01   .2776E+03   .1120E+05   .1252E+03   .1104E+05
.9099E+01   .2815E+03   .1156E+05   .1270E+03   .1140E+05
.9223E+01   .2854E+03   .1192E+05   .1288E+03   .1177E+05
.9347E+01   .2894E+03   .1230E+05   .1306E+03   .1214E+05
.9471E+01   .2933E+03   .1268E+05   .1325E+03   .1252E+05
.9595E+01   .2973E+03   .1307E+05   .1343E+03   .1291E+05
.9719E+01   .3012E+03   .1346E+05   .1361E+03   .1330E+05
.9843E+01   .3051E+03   .1386E+05   .1380E+03   .1370E+05
.9967E+01   .3091E+03   .1427E+05   .1398E+03   .1410E+05
.1007E+02   .3123E+03   .1462E+05   .1413E+03   .1445E+05
.1017E+02   .3156E+03   .1497E+05   .1429E+03   .1480E+05
.1027E+02   .3189E+03   .1532E+05   .1444E+03   .1515E+05
.1038E+02   .3221E+03   .1568E+05   .1459E+03   .1551E+05
.1048E+02   .3254E+03   .1605E+05   .1475E+03   .1587E+05
.1058E+02   .3286E+03   .1642E+05   .1490E+03   .1625E+05
.1068E+02   .3319E+03   .1681E+05   .1505E+03   .1663E+05
.1078E+02   .3352E+03   .1720E+05   .1520E+03   .1701E+05
.1089E+02   .3384E+03   .1760E+05   .1536E+03   .1741E+05
.1099E+02   .3417E+03   .1800E+05   .1551E+03   .1782E+05
.1109E+02   .3449E+03   .1842E+05   .1566E+03   .1823E+05
.1119E+02   .3482E+03   .1884E+05   .1581E+03   .1865E+05
.1129E+02   .3515E+03   .1928E+05   .1597E+03   .1908E+05
.1140E+02   .3547E+03   .1972E+05   .1612E+03   .1952E+05
.1150E+02   .3580E+03   .2017E+05   .1627E+03   .1997E+05
.1160E+02   .3613E+03   .2063E+05   .1643E+03   .2043E+05
.1170E+02   .3645E+03   .2110E+05   .1658E+03   .2090E+05
.1180E+02   .3678E+03   .2158E+05   .1673E+03   .2138E+05
.1191E+02   .3710E+03   .2208E+05   .1688E+03   .2187E+05
.1201E+02   .3743E+03   .2258E+05   .1704E+03   .2236E+05
.1211E+02   .3776E+03   .2310E+05   .1719E+03   .2288E+05
.1221E+02   .3808E+03   .2364E+05   .1734E+03   .2341E+05
.1231E+02   .3841E+03   .2419E+05   .1750E+03   .2398E+05
.1242E+02   .3873E+03   .2475E+05   .1765E+03   .2454E+05
.1252E+02   .3906E+03   .2532E+05   .1780E+03   .2511E+05
.1271E+02   .3968E+03   .2646E+05   .1809E+03   .2624E+05
.1291E+02   .4073E+03   .2765E+05   .1858E+03   .2743E+05
.1311E+02   .4177E+03   .2890E+05   .1907E+03   .2867E+05
```

```
.1330E+02    .4281E+03    .3021E+05    .1956E+03    .2998E+05
.1350E+02    .4386E+03    .3158E+05    .2005E+03    .3133E+05
.1369E+02    .4490E+03    .3302E+05    .2053E+03    .3277E+05
.1389E+02    .4594E+03    .3443E+05    .2102E+03    .3422E+05
.1408E+02    .4699E+03    .3603E+05    .2151E+03    .3577E+05
.1428E+02    .4803E+03    .3761E+05    .2200E+03    .3735E+05
.1448E+02    .4907E+03    .3925E+05    .2249E+03    .3899E+05
.1467E+02    .5012E+03    .4095E+05    .2298E+03    .4068E+05
.1487E+02    .5116E+03    .4272E+05    .2347E+03    .4244E+05
.1506E+02    .5220E+03    .4457E+05    .2396E+03    .4429E+05
.1526E+02    .5325E+03    .4651E+05    .2445E+03    .4622E+05
.1545E+02    .5429E+03    .4855E+05    .2494E+03    .4826E+05
.1565E+02    .5533E+03    .5069E+05    .2542E+03    .5039E+05
.1584E+02    .5638E+03    .5293E+05    .2591E+03    .5262E+05
.1604E+02    .5742E+03    .5528E+05    .2640E+03    .5497E+05
.1624E+02    .5846E+03    .5774E+05    .2689E+03    .5742E+05
.1643E+02    .5951E+03    .6032E+05    .2738E+03    .6000E+05
.1663E+02    .6055E+03    .6302E+05    .2787E+03    .6270E+05
.1682E+02    .6159E+03    .6585E+05    .2835E+03    .6552E+05
.1702E+02    .6264E+03    .6881E+05    .2884E+03    .6848E+05
.1721E+02    .6368E+03    .7191E+05    .2933E+03    .7157E+05
.1741E+02    .6472E+03    .7515E+05    .2982E+03    .7480E+05
.1752E+02    .6531E+03    .7707E+05    .3010E+03    .7672E+05
.1763E+02    .6611E+03    .7904E+05    .3047E+03    .7869E+05
.1774E+02    .6692E+03    .8107E+05    .3084E+03    .8071E+05
.1785E+02    .6772E+03    .8315E+05    .3122E+03    .8278E+05
.1796E+02    .6852E+03    .8528E+05    .3159E+03    .8491E+05
.1807E+02    .6932E+03    .8746E+05    .3197E+03    .8709E+05
.1818E+02    .7012E+03    .8970E+05    .3234E+03    .8933E+05
.1830E+02    .7092E+03    .9200E+05    .3272E+03    .9162E+05
.1841E+02    .7172E+03    .9435E+05    .3309E+03    .9397E+05
.1852E+02    .7253E+03    .9676E+05    .3347E+03    .9637E+05
.1863E+02    .7333E+03    .9922E+05    .3384E+03    .9882E+05
.1874E+02    .7413E+03    .1017E+06    .3422E+03    .1013E+06
.1885E+02    .7493E+03    .1043E+06    .3459E+03    .1039E+06
.1896E+02    .7573E+03    .1069E+06    .3497E+03    .1065E+06
.1907E+02    .7653E+03    .1095E+06    .3534E+03    .1091E+06
.1918E+02    .7733E+03    .1121E+06    .3572E+03    .1117E+06
.1929E+02    .7814E+03    .1148E+06    .3610E+03    .1144E+06
.1940E+02    .7894E+03    .1175E+06    .3647E+03    .1171E+06
.1951E+02    .7974E+03    .1203E+06    .3685E+03    .1198E+06
.1962E+02    .8054E+03    .1230E+06    .3723E+03    .1226E+06
.1973E+02    .8134E+03    .1258E+06    .3761E+03    .1254E+06
.1984E+02    .8214E+03    .1287E+06    .3799E+03    .1282E+06
.1995E+02    .8294E+03    .1315E+06    .3836E+03    .1311E+06
.2006E+02    .8374E+03    .1344E+06    .3874E+03    .1340E+06
```

Beach Sand (Plain Water)
Case C, Porosity = 0.40




Beach Sand (Plain Water)
Case C, Porosity = 0.40
Effective Stress (bars)
500
400
300
200
100
0
0
3
6
9
12
15
Axial Strain (%)
axial
radial

D.2.2 Cemented Coral

TAPE5 contains the effective axial strain-effective axial stress pairs and TAPE8 contains the effective axial strain-k_o pairs.

TAPE5

```
0.41          13             50             .20

0.0             0.0
 .00348       324.
 .040         365.
 .076         427.
 .132         526.
 .180         628.
 .206         721.
 .219         781.
 .241         917.
 .262        1082.
 .276        1255.
 .284        1366.
 .287        1434.
```

TAPE8

```
4

0.0              .1
 .00348          .75
 .157            .47
 .30             .47
```

TAPEs 3 and 6 contain the output data listed in subsection D.2.1.

List of Output File Tape 3

ε_{ai}	M_{fi} (bars)	π_i (bars)	n_i
.1467E-01	.1325E+06	.6488E+01	.4099E+00
.2935E-01	.1325E+06	.1298E+02	.4099E+00
.4402E-01	.1325E+06	.1946E+02	.4098E+00
.5869E-01	.1325E+06	.2595E+02	.4098E+00
.7336E-01	.1325E+06	.3244E+02	.4097E+00
.8804E-01	.1325E+06	.3893E+02	.4096E+00
.1027E+00	.1326E+06	.4542E+02	.4096E+00
.1174E+00	.1326E+06	.5191E+02	.4095E+00
.1321E+00	.1326E+06	.5840E+02	.4094E+00
.1467E+00	.1326E+06	.6491E+02	.4094E+00
.1614E+00	.1327E+06	.7142E+02	.4093E+00
.1761E+00	.1328E+06	.7796E+02	.4093E+00
.1908E+00	.1329E+06	.8450E+02	.4092E+00
.2054E+00	.1329E+06	.9106E+02	.4091E+00
.2201E+00	.1330E+06	.9763E+02	.4091E+00
.2348E+00	.1331E+06	.1042E+03	.4090E+00
.2495E+00	.1332E+06	.1108E+03	.4089E+00
.2642E+00	.1332E+06	.1174E+03	.4089E+00
.2789E+00	.1333E+06	.1240E+03	.4088E+00
.2936E+00	.1334E+06	.1307E+03	.4088E+00
.3082E+00	.1335E+06	.1373E+03	.4087E+00
.3229E+00	.1336E+06	.1440E+03	.4086E+00
.3376E+00	.1336E+06	.1507E+03	.4086E+00
.3523E+00	.1337E+06	.1574E+03	.4085E+00
.3670E+00	.1338E+06	.1641E+03	.4085E+00
.5333E+00	.5126E+05	.2476E+03	.4076E+00
.7001E+00	.5242E+05	.3334E+03	.4068E+00
.8675E+00	.5362E+05	.4215E+03	.4059E+00
.1035E+01	.5486E+05	.5120E+03	.4051E+00
.1204E+01	.5615E+05	.6050E+03	.4042E+00
.1373E+01	.5748E+05	.7006E+03	.4034E+00
.1543E+01	.5886E+05	.7990E+03	.4025E+00
.1714E+01	.6030E+05	.9002E+03	.4017E+00
.1885E+01	.6179E+05	.1004E+04	.4008E+00
.2057E+01	.6331E+05	.1112E+04	.3999E+00
.2229E+01	.6444E+05	.1221E+04	.3991E+00
.2401E+01	.6600E+05	.1333E+04	.3982E+00
.2571E+01	.6772E+05	.1446E+04	.3973E+00
.2740E+01	.6944E+05	.1562E+04	.3965E+00
.2908E+01	.7161E+05	.1680E+04	.3956E+00
.3075E+01	.7422E+05	.1803E+04	.3947E+00
.3242E+01	.7690E+05	.1929E+04	.3938E+00
.3407E+01	.7967E+05	.2060E+04	.3929E+00
.3573E+01	.8221E+05	.2194E+04	.3920E+00
.3738E+01	.8445E+05	.2332E+04	.3911E+00
.3902E+01	.8673E+05	.2473E+04	.3902E+00
.4065E+01	.8905E+05	.2616E+04	.3893E+00
.4228E+01	.9141E+05	.2764E+04	.3884E+00
.4391E+01	.9382E+05	.2915E+04	.3875E+00
.4553E+01	.9628E+05	.3069E+04	.3865E+00
.4713E+01	.9967E+05	.3226E+04	.3856E+00
.4872E+01	.1031E+06	.3388E+04	.3847E+00
.5031E+01	.1066E+06	.3555E+04	.3838E+00

```
.5190E+01    .1102E+06    .3727E+04    .3829E+00
.5349E+01    .1140E+06    .3906E+04    .3820E+00
.5507E+01    .1179E+06    .4090E+04    .3810E+00
.5666E+01    .1220E+06    .4281E+04    .3801E+00
.5824E+01    .1262E+06    .4478E+04    .3792E+00
.5982E+01    .1306E+06    .4683E+04    .3782E+00
.6140E+01    .1352E+06    .4894E+04    .3773E+00
.6298E+01    .1399E+06    .5112E+04    .3764E+00
.6456E+01    .1448E+06    .5338E+04    .3754E+00
.6614E+01    .1498E+06    .5572E+04    .3745E+00
.6771E+01    .1550E+06    .5814E+04    .3735E+00
.6929E+01    .1604E+06    .6064E+04    .3726E+00
.7086E+01    .1660E+06    .6323E+04    .3716E+00
.7243E+01    .1719E+06    .6591E+04    .3706E+00
.7401E+01    .1780E+06    .6868E+04    .3697E+00
.7558E+01    .1844E+06    .7156E+04    .3687E+00
.7715E+01    .1912E+06    .7454E+04    .3677E+00
.7872E+01    .1985E+06    .7763E+04    .3667E+00
.8029E+01    .2060E+06    .8084E+04    .3658E+00
.8186E+01    .2141E+06    .8418E+04    .3648E+00
.8344E+01    .2232E+06    .8767E+04    .3638E+00
.8501E+01    .2326E+06    .9130E+04    .3628E+00
.8745E+01    .2429E+06    .9720E+04    .3613E+00
.8990E+01    .2607E+06    .1035E+05    .3597E+00
.9235E+01    .2745E+06    .1102E+05    .3581E+00
.9479E+01    .2844E+06    .1171E+05    .3566E+00
.9722E+01    .2946E+06    .1242E+05    .3550E+00
.9965E+01    .3052E+06    .1316E+05    .3534E+00
.1021E+02    .3161E+06    .1393E+05    .3518E+00
.1045E+02    .3273E+06    .1471E+05    .3502E+00
.1069E+02    .3389E+06    .1553E+05    .3486E+00
.1093E+02    .3538E+06    .1638E+05    .3469E+00
.1118E+02    .3708E+06    .1727E+05    .3453E+00
.1142E+02    .3887E+06    .1821E+05    .3437E+00
.1166E+02    .4075E+06    .1919E+05    .3420E+00
.1190E+02    .4272E+06    .2021E+05    .3404E+00
.1214E+02    .4487E+06    .2129E+05    .3387E+00
.1238E+02    .4750E+06    .2243E+05    .3370E+00
.1262E+02    .5029E+06    .2365E+05    .3353E+00
.1286E+02    .5325E+06    .2493E+05    .3336E+00
.1311E+02    .5638E+06    .2629E+05    .3319E+00
.1335E+02    .5974E+06    .2773E+05    .3302E+00
.1359E+02    .6330E+06    .2925E+05    .3285E+00
.1383E+02    .6707E+06    .3087E+05    .3267E+00
.1407E+02    .7049E+06    .3257E+05    .3250E+00
.1432E+02    .7356E+06    .3435E+05    .3232E+00
.1456E+02    .7675E+06    .3620E+05    .3214E+00
.1476E+02    .8012E+06    .3785E+05    .3199E+00
.1497E+02    .8310E+06    .3957E+05    .3184E+00
.1518E+02    .8618E+06    .4134E+05    .3169E+00
.1538E+02    .9025E+06    .4320E+05    .3153E+00
.1559E+02    .9480E+06    .4516E+05    .3138E+00
.1580E+02    .9958E+06    .4722E+05    .3122E+00
.1600E+02    .1046E+07    .4938E+05    .3106E+00
.1621E+02    .1099E+07    .5165E+05    .3090E+00
```

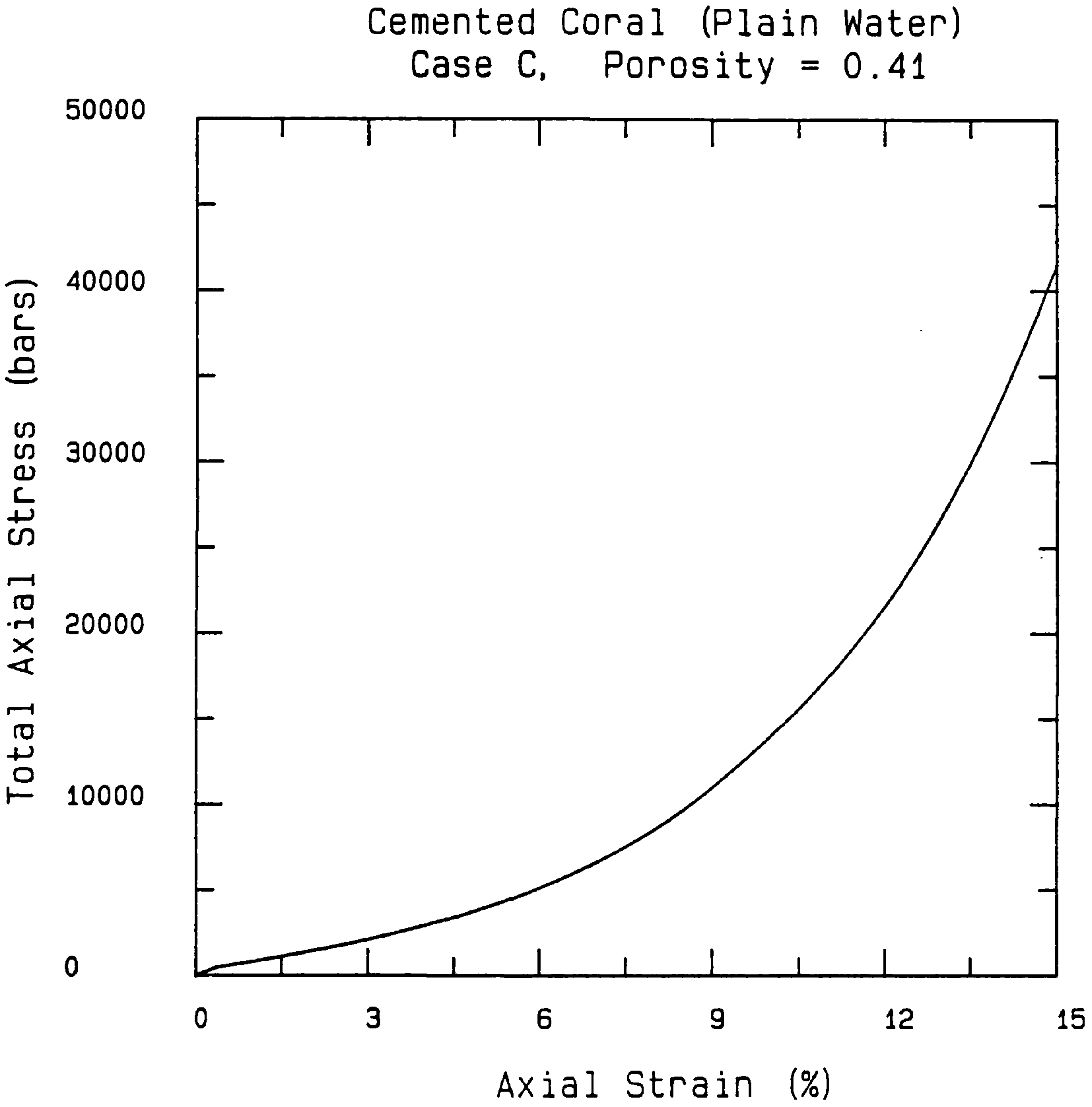
```
.1642E+02   .1154E+07   .5403E+05   .3075E+00
.1663E+02   .1211E+07   .5654E+05   .3059E+00
.1683E+02   .1271E+07   .5916E+05   .3043E+00
.1704E+02   .1334E+07   .6192E+05   .3027E+00
.1725E+02   .1398E+07   .6482E+05   .3010E+00
.1746E+02   .1465E+07   .6785E+05   .2994E+00
.1766E+02   .1535E+07   .7098E+05   .2978E+00
.1786E+02   .1608E+07   .7425E+05   .2962E+00
.1807E+02   .1685E+07   .7768E+05   .2946E+00
.1827E+02   .1766E+07   .8127E+05   .2930E+00
.1847E+02   .1850E+07   .8504E+05   .2913E+00
.1868E+02   .1935E+07   .8897E+05   .2897E+00
.1888E+02   .2023E+07   .9309E+05   .2880E+00
.1909E+02   .2108E+07   .9739E+05   .2864E+00
.1929E+02   .2194E+07   .1019E+06   .2847E+00
.1949E+02   .2266E+07   .1065E+06   .2830E+00
.1970E+02   .2316E+07   .1112E+06   .2813E+00
.1981E+02   .2367E+07   .1138E+06   .2804E+00
.1992E+02   .2395E+07   .1164E+06   .2795E+00
.2003E+02   .2423E+07   .1190E+06   .2786E+00
```

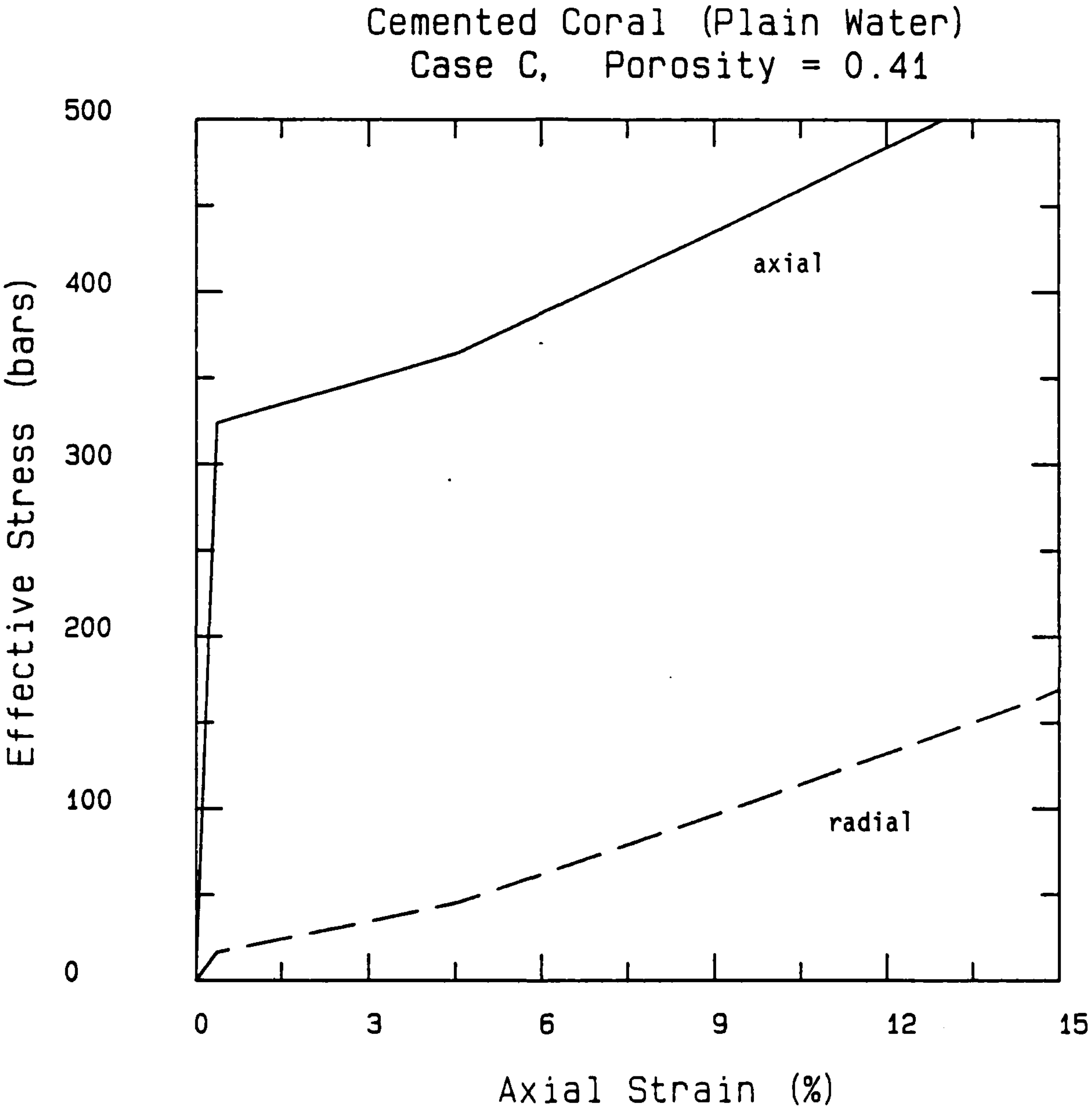
List of Output File Tape 6

ε_{ai}	σ'_{ai} (bars)	σ_{ai} (bars)	σ'_{ri} (bars)	σ_{ri} (bars)
.1467E-01	.1296E+02	.1945E+02	.6654E+00	.7153E+01
.2935E-01	.2592E+02	.3890E+02	.1331E+01	.1431E+02
.4402E-01	.3888E+02	.5834E+02	.1996E+01	.2146E+02
.5869E-01	.5184E+02	.7779E+02	.2661E+01	.2861E+02
.7336E-01	.6480E+02	.9724E+02	.3327E+01	.3577E+02
.8804E-01	.7776E+02	.1167E+03	.3992E+01	.4292E+02
.1027E+00	.9072E+02	.1361E+03	.4657E+01	.5008E+02
.1174E+00	.1037E+03	.1556E+03	.5323E+01	.5723E+02
.1321E+00	.1166E+03	.1750E+03	.5988E+01	.6439E+02
.1467E+00	.1296E+03	.1945E+03	.6651E+01	.7156E+02
.1614E+00	.1426E+03	.2140E+03	.7314E+01	.7874E+02
.1761E+00	.1555E+03	.2335E+03	.7975E+01	.8593E+02
.1908E+00	.1685E+03	.2530E+03	.8635E+01	.9313E+02
.2054E+00	.1814E+03	.2725E+03	.9293E+01	.1004E+03
.2201E+00	.1944E+03	.2920E+03	.9951E+01	.1076E+03
.2348E+00	.2074E+03	.3116E+03	.1061E+02	.1148E+03
.2495E+00	.2203E+03	.3311E+03	.1126E+02	.1221E+03
.2642E+00	.2333E+03	.3507E+03	.1192E+02	.1293E+03
.2789E+00	.2462E+03	.3703E+03	.1257E+02	.1366E+03
.2936E+00	.2592E+03	.3899E+03	.1322E+02	.1439E+03
.3082E+00	.2722E+03	.4095E+03	.1387E+02	.1512E+03
.3229E+00	.2851E+03	.4291E+03	.1452E+02	.1585E+03
.3376E+00	.2981E+03	.4487E+03	.1516E+02	.1658E+03
.3523E+00	.3110E+03	.4684E+03	.1581E+02	.1732E+03
.3670E+00	.3240E+03	.4881E+03	.1645E+02	.1805E+03
.5333E+00	.3256E+03	.5733E+03	.1763E+02	.2653E+03
.7001E+00	.3273E+03	.6607E+03	.1880E+02	.3522E+03
.8675E+00	.3289E+03	.7505E+03	.1997E+02	.4415E+03
.1035E+01	.3306E+03	.8426E+03	.2114E+02	.5332E+03
.1204E+01	.3322E+03	.9372E+03	.2230E+02	.6273E+03
.1373E+01	.3338E+03	.1034E+04	.2347E+02	.7241E+03
.1543E+01	.3355E+03	.1134E+04	.2463E+02	.8236E+03
.1714E+01	.3371E+03	.1237E+04	.2579E+02	.9260E+03
.1885E+01	.3388E+03	.1343E+04	.2695E+02	.1031E+04
.2057E+01	.3404E+03	.1452E+04	.2811E+02	.1140E+04
.2229E+01	.3420E+03	.1563E+04	.2927E+02	.1250E+04
.2401E+01	.3437E+03	.1676E+04	.3043E+02	.1363E+04
.2571E+01	.3453E+03	.1792E+04	.3159E+02	.1478E+04
.2740E+01	.3470E+03	.1909E+04	.3275E+02	.1595E+04
.2908E+01	.3486E+03	.2029E+04	.3392E+02	.1714E+04
.3075E+01	.3502E+03	.2153E+04	.3509E+02	.1838E+04
.3242E+01	.3519E+03	.2281E+04	.3627E+02	.1961E+04
.3407E+01	.3535E+03	.2413E+04	.3744E+02	.2097E+04
.3573E+01	.3552E+03	.2549E+04	.3862E+02	.2233E+04
.3738E+01	.3568E+03	.2688E+04	.3979E+02	.2371E+04
.3902E+01	.3584E+03	.2831E+04	.4097E+02	.2513E+04
.4065E+01	.3601E+03	.2976E+04	.4215E+02	.2659E+04

```
.4228E+01     .3617E+03     .3126E+04     .4334E+02     .2807E+04
.4391E+01     .3634E+03     .3278E+04     .4452E+02     .2959E+04
.4553E+01     .3650E+03     .3434E+04     .4570E+02     .3115E+04
.4713E+01     .3675E+03     .3593E+04     .4750E+02     .3273E+04
.4872E+01     .3700E+03     .3757E+04     .4929E+02     .3437E+04
.5031E+01     .3724E+03     .3927E+04     .5109E+02     .3606E+04
.5190E+01     .3749E+03     .4102E+04     .5288E+02     .3780E+04
.5349E+01     .3774E+03     .4283E+04     .5468E+02     .3961E+04
.5507E+01     .3799E+03     .4470E+04     .5648E+02     .4147E+04
.5666E+01     .3824E+03     .4663E+04     .5827E+02     .4339E+04
.5824E+01     .3848E+03     .4863E+04     .6007E+02     .4538E+04
.5982E+01     .3873E+03     .5070E+04     .6187E+02     .4744E+04
.6140E+01     .3898E+03     .5283E+04     .6367E+02     .4957E+04
.6298E+01     .3923E+03     .5504E+04     .6547E+02     .5178E+04
.6456E+01     .3948E+03     .5733E+04     .6727E+02     .5405E+04
.6614E+01     .3972E+03     .5969E+04     .6907E+02     .5641E+04
.6771E+01     .3997E+03     .6214E+04     .7087E+02     .5885E+04
.6929E+01     .4022E+03     .6466E+04     .7268E+02     .6137E+04
.7086E+01     .4047E+03     .6728E+04     .7448E+02     .6397E+04
.7243E+01     .4072E+03     .6998E+04     .7628E+02     .6667E+04
.7401E+01     .4096E+03     .7278E+04     .7808E+02     .6947E+04
.7558E+01     .4121E+03     .7568E+04     .7989E+02     .7236E+04
.7715E+01     .4146E+03     .7868E+04     .8169E+02     .7535E+04
.7872E+01     .4171E+03     .8180E+04     .8349E+02     .7847E+04
.8029E+01     .4196E+03     .8504E+04     .8530E+02     .8170E+04
.8186E+01     .4220E+03     .8840E+04     .8710E+02     .8505E+04
.8344E+01     .4245E+03     .9191E+04     .8890E+02     .8855E+04
.8501E+01     .4270E+03     .9557E+04     .9071E+02     .9220E+04
.8745E+01     .4310E+03     .1015E+05     .9359E+02     .9813E+04
.8990E+01     .4349E+03     .1079E+05     .9646E+02     .1045E+05
.9235E+01     .4389E+03     .1146E+05     .9934E+02     .1112E+05
.9479E+01     .4428E+03     .1215E+05     .1022E+03     .1181E+05
.9722E+01     .4468E+03     .1287E+05     .1051E+03     .1253E+05
.9965E+01     .4508E+03     .1361E+05     .1080E+03     .1327E+05
.1021E+02     .4547E+03     .1438E+05     .1109E+03     .1404E+05
.1045E+02     .4587E+03     .1517E+05     .1138E+03     .1483E+05
.1069E+02     .4626E+03     .1599E+05     .1167E+03     .1565E+05
.1093E+02     .4666E+03     .1685E+05     .1196E+03     .1650E+05
.1118E+02     .4706E+03     .1774E+05     .1224E+03     .1739E+05
.1142E+02     .4745E+03     .1868E+05     .1253E+03     .1833E+05
.1166E+02     .4785E+03     .1967E+05     .1282E+03     .1931E+05
.1190E+02     .4824E+03     .2070E+05     .1311E+03     .2034E+05
.1214E+02     .4864E+03     .2178E+05     .1340E+03     .2141E+05
.1238E+02     .4904E+03     .2292E+05     .1369E+03     .2257E+05
.1262E+02     .4943E+03     .2414E+05     .1398E+03     .2379E+05
.1286E+02     .4983E+03     .2543E+05     .1427E+03     .2507E+05
.1311E+02     .5022E+03     .2679E+05     .1456E+03     .2643E+05
.1335E+02     .5062E+03     .2823E+05     .1485E+03     .2788E+05
.1359E+02     .5102E+03     .2976E+05     .1514E+03     .2941E+05
```

```
.1383E+02    .5141E+03    .3139E+05    .1543E+03    .3103E+05
.1407E+02    .5181E+03    .3309E+05    .1572E+03    .3273E+05
.1432E+02    .5220E+03    .3487E+05    .1600E+03    .3451E+05
.1456E+02    .5260E+03    .3672E+05    .1629E+03    .3636E+05
.1476E+02    .5301E+03    .3838E+05    .1659E+03    .3802E+05
.1497E+02    .5342E+03    .4010E+05    .1689E+03    .3973E+05
.1518E+02    .5382E+03    .4188E+05    .1719E+03    .4151E+05
.1538E+02    .5423E+03    .4375E+05    .1749E+03    .4338E+05
.1559E+02    .5464E+03    .4571E+05    .1778E+03    .4534E+05
.1580E+02    .5505E+03    .4777E+05    .1808E+03    .4740E+05
.1600E+02    .5546E+03    .4993E+05    .1838E+03    .4956E+05
.1621E+02    .5586E+03    .5221E+05    .1868E+03    .5183E+05
.1642E+02    .5627E+03    .5459E+05    .1898E+03    .5422E+05
.1663E+02    .5668E+03    .5710E+05    .1927E+03    .5673E+05
.1683E+02    .5709E+03    .5974E+05    .1957E+03    .5936E+05
.1704E+02    .5750E+03    .6250E+05    .1987E+03    .6212E+05
.1725E+02    .5790E+03    .6540E+05    .2017E+03    .6502E+05
.1746E+02    .5831E+03    .6844E+05    .2047E+03    .6806E+05
.1766E+02    .5872E+03    .7156E+05    .2065E+03    .7118E+05
.1786E+02    .5913E+03    .7484E+05    .2082E+03    .7445E+05
.1807E+02    .5954E+03    .7827E+05    .2100E+03    .7789E+05
.1827E+02    .5994E+03    .8187E+05    .2118E+03    .8148E+05
.1847E+02    .6035E+03    .8564E+05    .2136E+03    .8525E+05
.1868E+02    .6076E+03    .8958E+05    .2154E+03    .8919E+05
.1888E+02    .6117E+03    .9371E+05    .2172E+03    .9331E+05
.1909E+02    .6158E+03    .9800E+05    .2189E+03    .9760E+05
.1929E+02    .6198E+03    .1025E+06    .2207E+03    .1021E+06
.1949E+02    .6239E+03    .1071E+06    .2225E+03    .1067E+06
.1970E+02    .6280E+03    .1118E+06    .2243E+03    .1114E+06
.1981E+02    .6317E+03    .1144E+06    .2260E+03    .1140E+06
.1992E+02    .6354E+03    .1170E+06    .2276E+03    .1166E+06
.2003E+02    .6392E+03    .1197E+06    .2292E+03    .1193E+06
```


Cemented Coral (Plain Water)
Case C, Porosity = 0.41
50000
40000
30000
20000
10000
0
Total Axial Stress (bars)
0
3
6
9
12
15
Axial Strain (%)



Cemented Coral (Plain Water)
Case C, Porosity = 0.41

APPENDIX E

VISCOUS FRICTION EQUATION INCORPORATING BIOT'S THEORETICAL AND WARD'S EMPIRICAL RESULTS

The fluid friction equation presented by Kim and Blouin (1984) is the generalized form of Biot's theoretical fluid equation. The friction force is proportional to the relative fluid velocity and the relative fluid acceleration.

Ward (1964) conducted a series of experimental flow tests and, based on analysis of these tests, proposed an empirical equation which is applicable for both laminar and turbulent flow in porous media. However, Ward's work was limited to steady state flow. The friction force in this equation is a function of both the relative fluid velocity and the square of the relative fluid velocity.

A generalized viscous friction equation combining our generalization of Biot's theoretical equation and Ward's empirical equation is given by:

$$\boxed{D_i = \frac{\gamma_f}{k}\dot{w}_i + \frac{\beta_f}{k^{1/2}}(\dot{w}_i)\dot{w}_i + \frac{\rho_f}{n}r\,\ddot{w}_i} \qquad \text{(E-1)}$$

where

D_i = viscous friction force between the pore fluid and the soil skeleton per unit volume of pore fluid

γ_f = unit weight of the pore fluid

ρ_f = mass density of the pore fluid

n = porosity

k = coefficient of permeability

r = mass increment factor

β_f = Ward's constant which depends on fluid properties

$\dot{w}_i$ = apparent relative fluid velocity

$\ddot{w}_i$ = apparent relative fluid acceleration

This equation requires validation by laboratory experiments and will be updated once non-steady-state flow experiments are conducted and analyzed.

APPENDIX F

ASSUMPTIONS ON TIME VARIATION OF FLUID PRESSURE

Figure F.1 shows the time variation of fluid pressure during the time step. The second derivative of fluid pressure with respect to time is assumed constant as in Newmark's average acceleration method.

$$\ddot{\bar{\pi}}_{\tau} = \ddot{\bar{\pi}}_{n-\frac{1}{2}} = \text{Constant} \qquad \text{(F-1)}$$

where τ varies from 0 to Δt.

Integrating Equation F-1 with respect to time,

$$\dot{\bar{\pi}}_{\tau} = \dot{\bar{\pi}}_{n-1} + \ddot{\bar{\pi}}_{n-\frac{1}{2}}\,\tau \qquad \text{(F-2)}$$

At $\tau = \frac{\Delta t}{2}$

$$\dot{\bar{\pi}}_{n-\frac{1}{2}} = \dot{\bar{\pi}}_{n-1} + \frac{\Delta t}{2}\,\ddot{\bar{\pi}}_{n-\frac{1}{2}} \qquad \text{(F-3)}$$

Since

$$\ddot{\bar{\pi}}_{n-\frac{1}{2}} = \tfrac{1}{2}(\ddot{\bar{\pi}}_{n-1} + \ddot{\bar{\pi}}_{n}) \qquad \text{(F-4)}$$

Substituting Equation F-4 into Equation F-3

$$\boxed{\dot{\bar{\pi}}_{n-\frac{1}{2}} = \dot{\bar{\pi}}_{n-1} + \frac{\Delta t}{4}\,\ddot{\bar{\pi}}_{n-1} + \frac{\Delta t}{4}\,\ddot{\bar{\pi}}_{n}} \qquad \text{(F-5)}$$

At $\tau = \Delta t$,

$$\boxed{\dot{\bar{\pi}}_n = \dot{\bar{\pi}}_{n-1} + \frac{\Delta t}{2}\left(\ddot{\bar{\pi}}_{n-1} + \ddot{\bar{\pi}}_n\right)} \qquad \text{(F-6)}$$

Integrating Equation F-2 with respect to time,

$$\bar{\pi}_\tau = \bar{\pi}_{n-1} + \dot{\bar{\pi}}_{n-1}\,\tau + \frac{\tau^2}{2}\,\ddot{\bar{\pi}}_{n-½} \qquad \text{(F-7)}$$

At $\tau = \Delta t$,

$$\bar{\pi}_n = \bar{\pi}_{n-1} + \dot{\bar{\pi}}_{n-1}\,\Delta t + \frac{\Delta t^2}{4}\,\ddot{\bar{\pi}}_{n-1} + \frac{\Delta t^2}{4}\,\ddot{\bar{\pi}}_n \qquad \text{(F-8)}$$

Further integrating Equation F-7 with respect to time,

$$\boxed{\int_{t_{n-1}}^{t_n} \bar{\pi}_{\tau'}\,d\tau' = \bar{\pi}_{n-1}\,\Delta t + \frac{\Delta t^2}{2}\,\dot{\bar{\pi}}_{n-1} + \frac{\Delta t^3}{12}\,\bar{\pi}_{n-1} + \frac{\Delta t^3}{12}\,\ddot{\bar{\pi}}_n} \qquad \text{(F-9)}$$

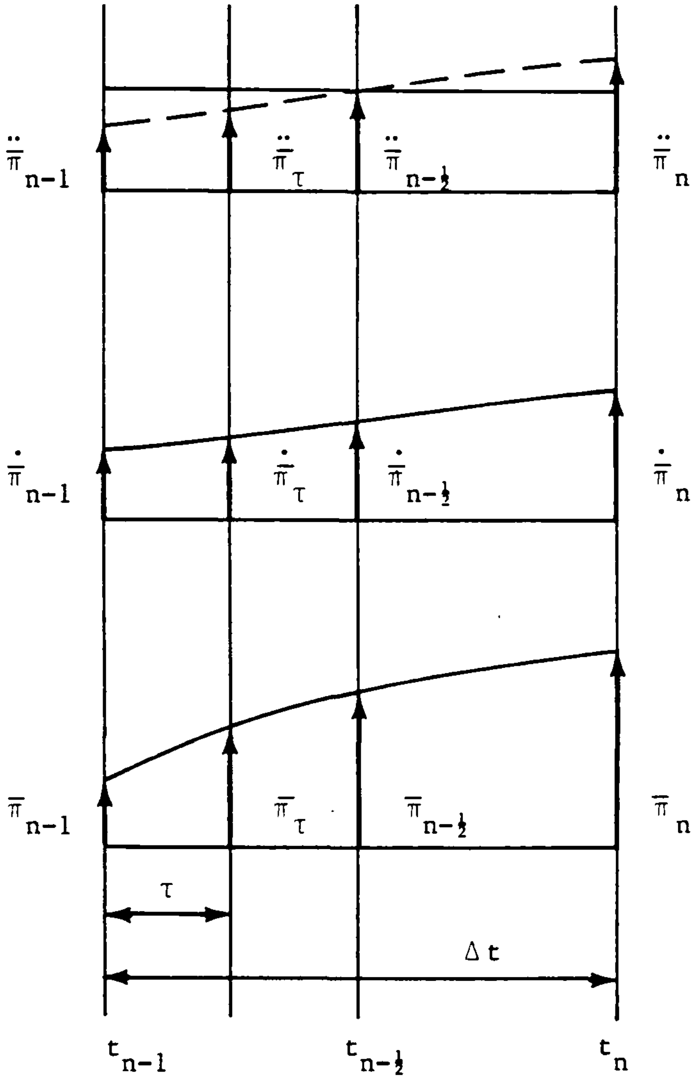


Figure F.1. Time variation of fluid pressure.

APPENDIX G

TPDAP II
User's Manual

TPDAP II

(Two Phase Dynamic Analysis Program)

December 1985

Kwang J. Kim

Applied Research Associates, Inc.
6020 Richmond Highway, Suite 204
Alexandria, VA 22303

CARD 1	Main Title	(80 characters)
CARD 2	Subtitle	(80 characters)

CARD 3 General Options

NF, NTCSF, ISFG, IP, NLNR, ICONST, NFG
(7I5)

Analysis Type
NF = 1 Static (available soon)
= 2 Consolidation (available soon)
= 3 Two phase dynamic

Options in Two Phase Dynamic Analysis (NF=3)
NTCSF = 0 Two phase dynamic analysis
1 Consolidation Analysis (suppressed inertial terms)
2 One phase solid dynamic analysis (suppressed fluid phase)
3 One phase fluid dynamic analysis (suppressed solid phase, available soon)

Initial Stress Conditions
ISFG = 0 No initial stress
1 specified effective stress and pore pressure
2 specified effective stress
3 specified pore pressure
4 imposed excess pore pressure

Stress and Strain Conditions
IP = 1 Plane stress
2 Plane strain
3 Axial symmetry
4 Spherical symmetry

Material Model
NLNR = 0 Linear elastic material model
1 Decoupled elastoplastic material model (DCOUP)
2 Cap model
3 AFWL engineering model
4 Uniaxial strain model (UNIAX)
5 ARA2D Model

Mass Distribution Options
ICONST= 0 Lumped mass (diagonal mass matrix)
1 Consistent mass (complete mass matrix, see J.S. Archer, "Consistent Mass Matrix for Distributed Systems," Proc. ASCE 89, ST4, 161, 1963)

Loading Type
NFG = 1 Specified base accelerations (available for earthquake analysis)
2 Specified pressure time history
3 Specified velocity time history
4 Both pressure and velocity time histories

CARD 4 Global Calculation Parameters

NCYCL, DT, NUPDAT, ITER
(I5, F10.0, 2I5)

NCYCL: Number of cycles (total number of time steps)
DT: Global time step (duration of each cycle)
NUPDAT: Number of cycles between updates to global stiffness matrix (ITER = 0 for NUPDAT>1)
ITER: Number of updates of global stiffness matrix within each cycle (NUPDAT = 1)

CARD 5 Mesh, Material and Boundary Parameters

NUMNP, NUMEL, NUMMAT, NVIS, NSKEW
(5I5)

NUMNP: Number of nodes
NUMEL: Number of elements
NUMMAT: Number of different materials
NVIS: Number of nodes on transmitting boundaries
NSKEW: Number of element sides on skew boundaries

CARD 6 Loading Functions

A. Acceleration Loading Functions

If NFG = 1, otherwise go to next card

MTYPE, NUMAP, DTXA (2I5, F10.0)

MTYPE = 0 Constant time increments in acceleration time history
1 Specified times in acceleration time history
NUMAP: Number of acceleration time pairs in the input acceleration time history
DTXA: Constant time increment in the input acceleration time history (for MTYPE = 0)

B. Pressure Loading Functions

If NFG = 2 or 4, otherwise go to next card

NUMLP, NUMLH, NUMTP, NTYPE, DTXX
(4I5, F10.0)

NUMLP: Total number of nodes at which input pressure time hisory is specified
NUMLH: Number of input pressure time histories
NUMTP: Number of pressure time pairs in every input pressure time history
NTYPE = 0 Constant time increments in pressure time history
1 Specified times in pressure time history
DTXX: Constant time increment in the input pressure time history (for NTYPE = 0)

CARD 7 Velocity Loading Functions

If NFG = 3 or 4, otherwise go to next card

NUMVEL, NUMVH, NUMVTP, NVTYPE, DTXV
(4I5, F10.0)

NUMVEL: Total number of nodes at which input velocity is specified
NUMVH: Number of input velocity time histories
NUMVTP: Number of velocity time pairs in every velocity time history
NVTYPE = 0 Constant time increments in velocity time
1 Specified times in velocity time history
DTXV: Constant time increment in the input velocity time history (for NVTYPE = 0)

CARD 8 Parameters Required For Storage Use

IVMDK, IELCEN, MAXIP, MXSH
(4I5)

IVMDK = 0 Use disk as sequential tape (good for CYBER)
= 1 Use disk as virtual memory
IELCEN = 0 Compute stresses at integration points
1 Compute stresses at element center
MAXIP: Maximum number of integration points per element
MXSH: Maximum number of stress/strain history data

NOTE: ---Stresses and pore pressures are calculated at element integration points or at element centers
---Motions (accelerations, velocities, and displacements) are calculated at nodes

CARD 9 Output Stress and/or Motion Profile Specifications

A. NPFL, NDC, NSG, NPRINT, NPEL, NPMT
(6I5)

NPFL = 0 Both motions at all nodes and stresses at all elements
1 Motions and stresses at specified nodes and elements respectively
2 Motions at all nodes
3 Stresses at specified elements
4 Stresses at all elements

NDC = 0 Write stress/displacement profile output to hard disc
1 Write stress/displacement profile output to floppy disc

NSG = 0 Write stress/displacement profile output in one file
1 Write stress/displacement profile output in specified files

NPRINT: Number of cycles between each output profile

NPEL: Number of elements in output stress profile

NPMT: Number of nodes in output motion profile

NOTE: If NDC = 1 and NSG = 0, the program writes displacement/stress profiles on hard disk 17 under the name "DISTR"

B. If NPFL = 1 or 3, otherwise go to Card 9c.

NPRT(I), I = 1, NPEL (free format)

NPRT(I); List specified element numbers in sequential order

C. If NPFL = 1, otherwise go to next card

NPM(I), I = 1, NPMT (free format)

NPM(I): List specified node numbers in sequential order

CARD 10 Output Stress and/or Motion Time History Specifications

A. NTHS, NHPEL, NHPMT (3I5)

Time history options

NTHS = 0 Do not print time history data, go to next card
1 Motion time histories
2 Stress time histories
3 Both motion and stress time histories

NHPEL: Number of elements at which stress time histories are required

NHPMT: Number of nodes at which motion time histories are required

B. If NTHS = 2 or 3, otherwise go to 10C.

NHPRT(I), I = 1, NHPEL (free format)

NHPRT(I): List specified element numbers in sequential order

C. If NTHS = 1 or 3, otherwise go to next card

NHPM(I), I = 1, NHPMT (free format)

NHPM(I): List specified node numbers in sequential order

CARD 11 Numerical Time-Integration Options

TETA, BETA, GAMA, ALPA, (4F10.0)

TETA: θ
BETA: β Refer to Table 1.
GAMA: γ
ALPA: α

TABLE 1. VALUES OF β AND θ FOR $\gamma = 1/2$* ($\alpha = 0$ By Default)]

Integration Method (1)	β (2)	θ (3)
Explicit second central difference	0	1.0
Fox-Goodwin	1/12	1.0
Linear Acceleration	1/6	1.0
Newmark's constant acceleration	1/4	1.0
Wilson	1/6	2.0
Stiff linear acceleration	1/6	1.5

* $\gamma = 1/2$ indicates no damping

$\gamma > 1/2$ introduces numerical damping and $\beta = (\gamma + 1/2)^2/4$

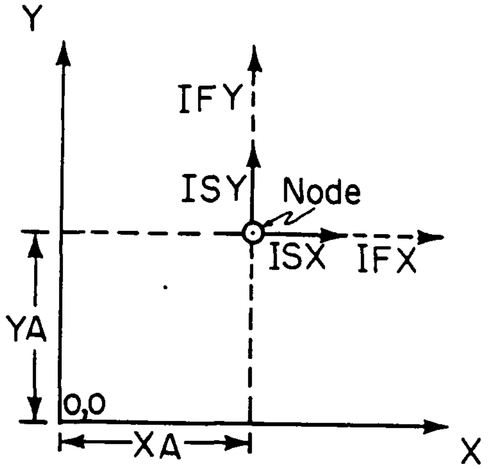
α Method available soon

For more information see: Ghaboussi and Wilson, "Variational Formulation of Dynamics of Fluid Saturated Porous Elastic Solids," ASCE Engineering Mechanics Journal, August 1972.

CARD 12 Nodal Coordinates and Degree of Freedom Specifications

For each node:

NODE, ISX, ISY, IFX, IFY, XA, YA
(5I5, 2F10.0)



NODE: Node Number
ISX: Specifies skeleton X Degrees of Freedom (DOF)
ISY: Specifies skeleton Y DOF
IFX: Specifies X DOF for relative pore fluid motion
IFY: Specifies Y DOF for relative pore fluid motion

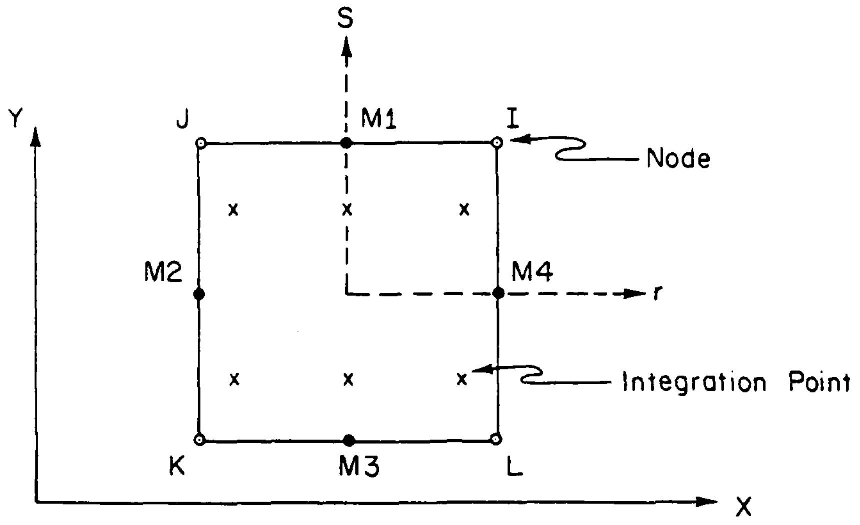
ISX, ISY, IFX, IFY = 0 Free to move in specified direction
= 1 Fixed in specified direction

XA: X Coordinate
YA: Y Coordinate: Note for IP = 4 (1-D spherical analysis) set the mesh height equal to 1.0. (see example spherical problem)

CARD 13 Element Specifications

For each element;

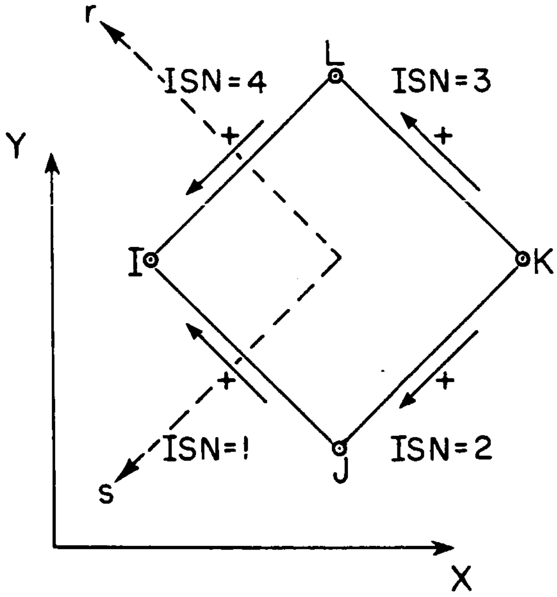
NEL, MAT, KS, KF, INTR, INTS, I, J, K, L, M1, M2, M3, M4
(14I5)



NEL:	Element number
MAT:	Material property number
KS = 0	Element has solid phase
1	Element has no solid phase
KF = 0	Element has fluid phase
1	Element has no fluid phase
INTR:	Number of integration points in r-direction
INTS:	Number of integration points in s-direction
I,J,K,L:	Node numbers at element corner
M1,M2,M3,M4:	Node numbers at element midside

CARD 14 Specification of Skew Boundaries

For each element side on a skew boundary:
NEL, NDT, NDH, ISN, MSF (5I5)



Note: 1. Positive directions on skew boundaries run parallel to element boundaries, from node K toward node I.
2. Element sides are numbered counterclockwise from node I
3. Input forces or velocities and output motions on skew boundaries are specified parallel to the element boundaries

NEL:	Element number
ISN:	Side number on skew boundary
NDT:	Node number at tail of arrow
NDH:	Node number at head of arrow
MSF = 1	Skew in fluid phase only
2	Skew in solid phase only
3	Skew in solid and fluid phases

NOTE: It is allowed only one skew boundary in current version.

CARD 15 Material Model Specifications and Parameters

A. ELS, TENS, STIFAC, SHEFAC, PMN, BK, G, D8
(8E10.3)

ELS = 0.0 Linear elastic material model
≠ 0.0 Material property number

TENS = 0.0 No tension cutoff
1.0 Tension cutoff (effective for NLNR ≠ 0)

STIFAC: Factor which reduces normal stiffness once tensile strength is exceeded (example: Reduced modulus = original modulus/STIFAC)

SHEFAC: Factor which reduces shear stiffness once shear strength is exceeded (see example above)

PMN: Tensile strength (tensile stress is positive)

BK: Elastic bulk modulus of skeleton

G: Elastic shear modulus of skeleton

D8: For NLNR = 1, G_p: Plastic shear modulus
For NLNR = 4, POSNR: Poisson's Ratio
For NLNR = 5, n: Material constant
Otherwise, leave blank

B. D9, D10, D11, D12, D13, D14, D15, D16
(8E10.3)

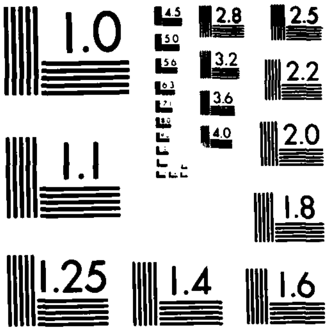
$$\left.\begin{array}{l}\text{For NLNR} = 1\\ \text{D9} = a\\ \text{D10} = b\end{array}\right\}\quad \text{Refer to Model DCOUP}$$

$$\left.\begin{array}{l}\text{For NLNR} = 4\\ \text{D9} = \text{EQNO}\\ \text{D10} = \text{C}\\ \text{D11} = \text{D}\\ \text{D12} = \text{SVMLL}\end{array}\right\}\quad \text{Refer to Model UNIAX}$$

$$\left.\begin{array}{l}\text{For NLNR} = 5\\ \text{D9} = \alpha\\ \text{D10} = \beta\\ \text{D11} = \kappa\\ \text{D12} = K\end{array}\right\}\quad \text{Refer to Model ARA2D}$$

Otherwise, leave blank

AD-A174 749   EXPERIMENTAL AND THEORETICAL RESPONSE OF MULTIPHASE   3/4
POROUS MEDIA TO DYNAM (U) APPLIED RESEARCH ASSOCIATES
INC SOUTH ROYALTON VT NEW ENGLAND   K J KIM ET AL
UNCLASSIFIED   13 AUG 86 ARA-5967-86 AFOSR-TR-86-0665   F/G 8/13   NL


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

C. D17; D18, RO, ROF, D21, D22, D23, D24
(8E10.3)

RO: Mass density of bulk mixture (ρ)
ROF: Mass density of pore fluid (ρ_f)

D. D25, CM, POR, RXX, RYY, RXY (6E10.3)

CM: Undrained bulk modulus of soil/water mixture

$$K_m = \frac{K_g K_w}{K_w + n(K_g - K_w)}$$

where K_g is bulk modulus of solid grains

K_w is bulk modulus of pore water

n is porosity

(Reference: Blouin and Kim, "Undrained Compressibility of Saturated Soil, Applied Research Associates, Inc., South Royalton, Vermont, 1984)

POR: Term in Biot's equation of motion for pore fluid which is a function of porosity and pore shape, POR = n/(1 + r)

r = 0, Generalized Darcy's Law (assumes steady state pore water flow or simplification of Biot's equation for non steady flow conditions)

r = 1/5, Biot's equation for flat pores (1961 paper)

r = 1/3, Biot's equation for circular pores (1961)

Other values determined experimentally or from parameter studies

RXX: Flow resistance in x direction (for isotropic permeability, $R_{xx} = \gamma_w/K$, where γ_w is unit weight of pore fluid and K is permeability in x direction)

RYY: Flow resistance in Y direction (For isotropic permeability, $R_{yy} = \gamma_w/K$)

RXY: Flow resistance term for general anisotropic permeability (Reference: Scheidegger, A.E., "The Physics of Flow Through Porous Media," Macmillan, New York, 1957.

CARD 16 Decoupled Elastoplastic Material Model (DCOUP)

If NLNR = 1, otherwise go to next card group.

A. NUMAT (I5)

NUMAT: Number of different material property sets.

For each material property set, provide input data describing volumetric and deviatoric behaviors.

DESCRIBE VOLUMETRIC BEHAVIOR

B.1 NLPC (I5)

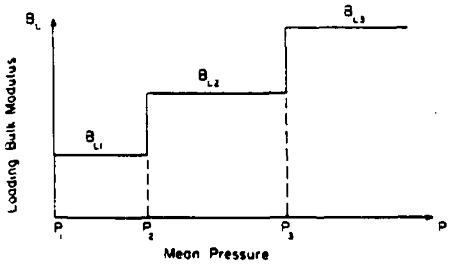
B.2 P_1, B_{L1}
P_2, B_{L2}
- -
- -
P_n B_{Ln}

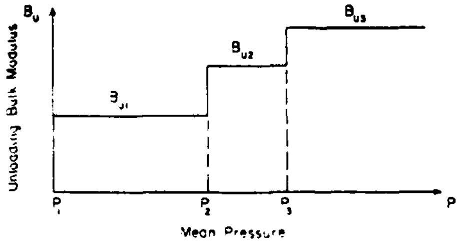
"NLPC" cards with format (2F10.0) for each card

B.3 NUPC (I5)

B.4 P_1, B_{U1}
P_2, B_{U2}
- -
- -
P_n, B_{Un}

"NUPC" cards with format (2F10.0) for each card



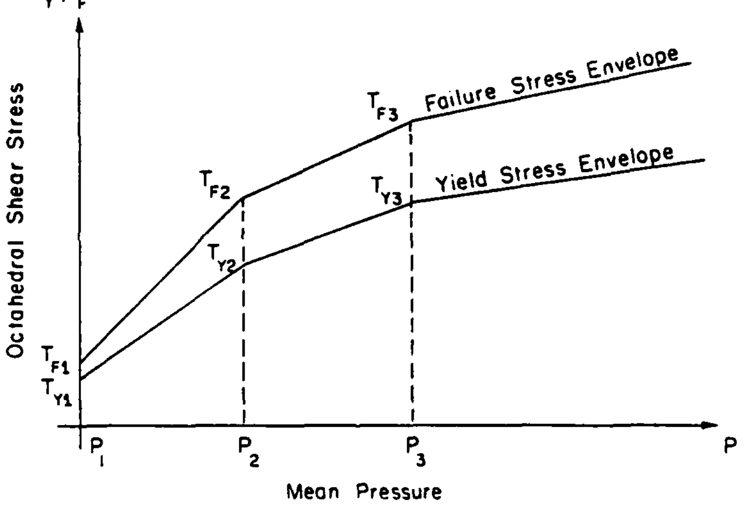


NLPC: Number of oressure/modulus pairs describing the loading bulk modulus
B_{Li}: Loading bulk modulus at mean pressure P_i
NUPC: Number of pressure/modulus pairs describing the unloading bulk modulus
B_{Ui}: Unloading bulk modulus at mean pressure P_i

DESCRIBE DEVIATORIC BEHAVIOR

C.1 NTPC (I5)

$$\text{C.2}\quad\left.\begin{array}{lll} P_1, & T_{Y1}, & T_{F1} \\ P_2, & T_{Y2}, & T_{F2} \\ - & - & - \\ \\ - & - & - \\ P_n, & P_{Yn}, & P_{Fn} \end{array}\right\}\begin{array}{l}\text{"NTPC" Cards} \\ \text{with format (3F10.0) for each card}\end{array}$$



NTPC: Number of pressure/yield stress/failure stress points describing the yield and failure envelopes as a function of mean pressure

T_{yi}, T_{Fi}: Octahedral shear stress at yield and failure respectively at mean pressure P_i

CARD 17 Uniaxial Strain Model (UNIAX)

If NLNR = 4, otherwise go to next card group

A. NUMAT (I5)

NUMAT: Number of different material property sets.

For each material property set, provide uniaxial loading and unloading behaviors.

B.1 NLPC (I5)

B.2

$$\left.\begin{array}{ll} S_1, & M_{L1} \\ - & - \\ - & - \\ S_n, & M_{Ln} \end{array}\right\} \text{"NLPC" cards with format (2F10.0) for each card}$$

B.3 NUPC (I5)

B.4

$$\left.\begin{array}{ll} S_1, & M_{U1} \\ - & - \\ - & - \\ S_n, & M_{Un} \end{array}\right\} \text{"NUPC" cards with format (2F10.0) for each card}$$

NLPC: Number of vertical stress-loading constrained modulus pairs

M_{Li}: Loading constrained modulus at vertical stress S_i

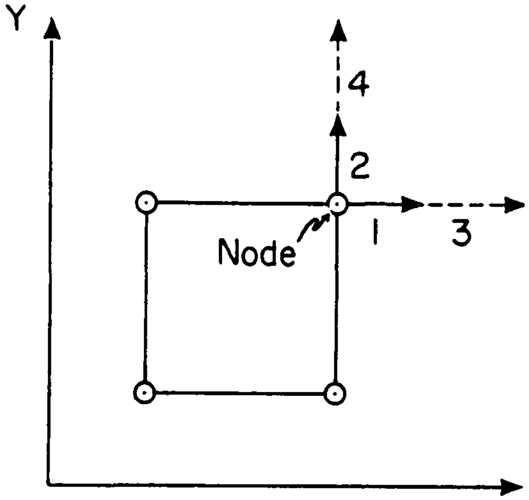
NUPC: Number of vertical stress-unloading constrained modulus pairs

M_{Ui}: Unloading constrained modulus at vertical stress S_i

CARD 18 Specifications of Transmitting Boundaries

HEADER (80 characters)

For each node on a transmitting boundary:
INOD, IDIR, VISC, (2I5,F10.0)



INOD: Node Number

IDIR: Specifies direction and phase of damping

IDIR = 1 Solid phase damping in x direction
2 Solid phase damping in y direction
3 Fluid phase damping in x direction
4 Fluid phase damping in y direction

VISC: Constant which is proportional to the force on a given node (ρcA_c), equal to impedence times contributory area on the node.

C equals C_p for IDIR normal to the transmitting boundary

C equals C_s for IDIR parallel to the transmitting boundary

Reference: Lysmer and Kuhlemeyer, "Finite Dynamic Model for Infinite Media," ASCE, EM4, pp. 859-877, 1969.

CARD 19 Initial In Situ Effective Stress Conditions

If ISFG from Card 3 equals 1 or 2, otherwise go to next card
For each element and for each stress point,
SXX, SYY, SZZ, SXY (4F10.0)

Initial effective skeleton stresses

SXX = σ_x' (normal stress in x direction)

SYY = σ_y' (normal stress in y direction)

SZZ = σ_z' (normal stress in z direction)

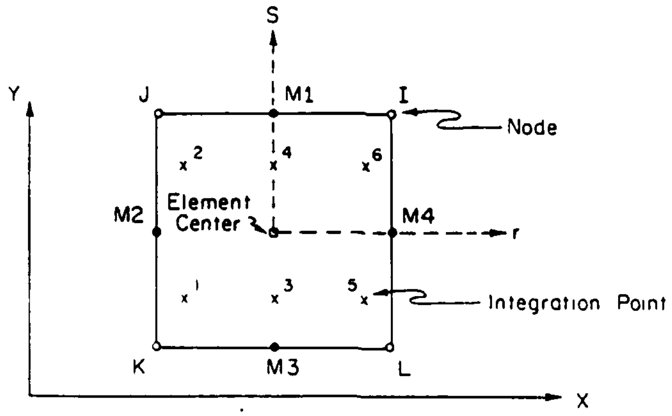
SXY = τ_{xy} (shear stress in xy plane)

CARD 20 Initial Pore Fluid Pressures

If ISFG from Card 3 equals 1, 3, or 4, otherwise go to next card

(PRF(J,I), J = 1, MXIP) I = 1, NUMEL (8F10.0)

PRF: List of initial pore fluid pressures in each element and at each stress point, specified sequentially from 1 to NUMEL. MXIP is the number of stress points



Note that tensile stresses are positive, the stress points are at integration points if IELCEN = 0 from Card 8 and are at element center if IELCEN = 1, and the sequence of integration points in element natural coordinate (r-s) is from bottom to top and left to right as shown in above figure.

CARD 21 Input Loading: for Specified Base Acceleration Time History (Card 3, NFG = 1)

A. Header (80 characters)

B. TSCAL, ASCAL (2E10.0)

C. If MTYPE = 1,
TS(I), GACL(I) (12F6.0) for I = 1, NUMAP
If MTYPE = 0,
GACL(I) (8F9.0) for I = 1, NUMAP

TSCAL: Time scaling factor used to extrapolate available ground motion data

ASCAL: Acceleration scaling factor used to extrapolate available ground motion data

TS(I): Specified times in the acceleration time history

GACL(I): Specified ground accelerations at each corresponding time TS(I)

CARD 22 Input Loading: For Specified Pressure Time Histories (Card 3, NFG = 2)

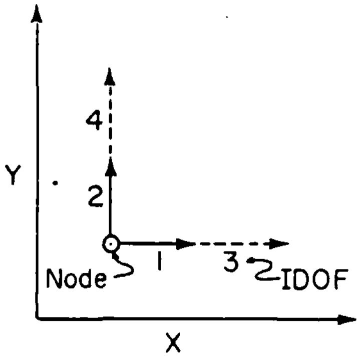
A. Header (80 characters)

B. For each of the NUMLP nodes (Card 6B) at which a load is applied:
NODE, IDOF, LHNO, CINT (3I5,E10.3)

C. TD(I), I = 1, NUMTP (7E10.0) for (NTYPE = 1)

D. For each of the NUMLH (Card 6B) loading time histories:
DYL(I), I = 1, NUMTP (7E10.0)

E. Comment Card (80 characters) - to separate loadings



NODE: Node number

IDOF = 1 Total force acting on a given node in the x direction

2 Total force acting on a given node in the y direction

3 Force acting on the pore fluid in the x direction at the given node (note: force on the fluid is part of the total force acting in the x direction)

4 Force acting on the pore fluid in the y direction

LHNO: Load history number (one of NUMLH load histories)

CINT: Load intensity factor (used to convert pressure or stress into force on a given node, based on contributing area)

TD(I): Set of specified times used in all the pressure loading histories

DYL(I): A set of pressure magnitudes input for each loading history at corresponding times TD(I)

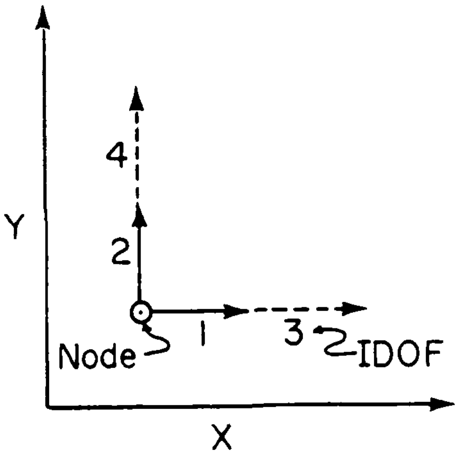
CARD 23 Input Loading for Specified Velocity Time Histories
(Card 3, NFG = 3 or 4)

A. Header (80 characters)

B. For each of the NUMVEL (card 7) nodes at which velocity is applied;
NODE, IDOF, LHNO, CINT (3I5,F10.0)

C. TV(I), I = 1, NUMVTP (7E10.0) (for NVTYPE = 1)

D. For each of the NUMVH (Card 7) input velocity time histories:
SVEL(I), I = 1, NUMVTP (7E10.0)



NODE: Node number

IDOF = 1 Skeleton motion in the x direction

2 Skeleton motion in the y direction

3 Apparent fluid motion in the x direction relative to the skeleton motion in the x direction (apparent displacement = n (V_x - U_x) where n is the porosity, V_x is the absolute fluid displacement in the x direction and U_x is the absolute skeleton displacemeint in the x direction)

4 Apparent fluid motion in the y direction (see above definition)

LHNO: Velocity history number (one of NUMVH velocity histories)

CINT: Velocity magnification factor used to change the velocity amplitude for a given velocity input (CINT * SVEL(I) gives velocity acting on the node of interest)

TV(I): Set of specified times used in all the velocity input histories

SVEL(I): A set of velocity magnitudes input for each loading history at corresponding times TD(I)

APPENDIX H

Source Listing of Program TPDAP II

```
C FTN7X,L
C $FILES(0,12)
C C $EMA /WAREA/,/ZAREA/,/MDCPL/,/TENCT/,/OUTPS/,/OUTHS/
C     PROGRAM TPDAP
      PROGRAM TPDAP(TAPE5,TAPE6,TAPE7,TAPE30,TAPE40,TAPE60,
     .              TAPE61,TAPE62,TAPE63,TAPE64,TAPE70,TAPE71,
     .              TAPE72,TAPE73,TAPE74,TAPE3,TAPE4,TAPE10,TAPE11,
     .              INPUT,OUTPUT)
C     INTEGER*4 N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15,
C    *  N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,N29,N30
C     INTEGER*4 N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B,NLAST
      COMMON /WAREA/ A(60000)
      COMMON /TENCT/ TCUT(4,4),HCUT(4,4)
      COMMON /NPARM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,
     * N14,N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,
     * N29,N30,N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B
      COMMON /EDT/ ALL(58)
      COMMON /INTCN/ AA(15),TETA,BETA,GAMA,ALPA
      COMMON /ZAREA/ Z(1500)
      COMMON /LPARM/ L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,MAXIP,MXIP,
     * MXSH,LBBB,LDDD,NELS
      COMMON /MATPR/ EMP(7),A2,B2,C2
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      COMMON /GCALP/ NCYCL,DT,NUPDAT,ITER
      COMMON /MMBDP/ NUMNP,NUMEL,NUMMAT,NVIS,NSKEW,NEQ,MBAND
      COMMON /ACCLF/ MTYPE,NUMAP,DTXA
      COMMON /PRELF/ NUMLP,NUMLH,NUMTP,NTYPE,DTXX
      COMMON /VELLF/ NUMVEL,NUMVH,NUMVTP,NVTYPE,DTXV
      COMMON /OUTPS/ NPFL,NDC,NSG,NPRINT,NPEL,NPMT,NPRT(1500),NPM(1500)
      COMMON /OUTHS/ NTHS,NHPEL,NHPMT,NHPRT(100),NHPM(100)
C     COMMON /NALLO/ NSDFL(8,9,2,2)
C     INTEGER LSEG(3),ISEG(3),JSEG(3),KSEG(3),LBUF(300),MSEG(3)
C     DATA LSEG/'DREAD'/
C     DATA ISEG/'DPSM1'/
C     DATA MSEG/'DPSMA'/
C     DATA JSEG/'DPSM2'/
C     DATA KSEG/'DPSM3'/
C     CALL LGBUF(LBUF,300)
C
C     CALL SEGLD(LSEG,IRO)
      CALL DREAD
C
      N1 = 1
      N2 = N1+4*NUMNP
      N3 = N2+NUMNP
      N4 = N3+NUMNP
      N5 = N4+30*NUMMAT
      N5A = N5
      N5B = N5
      IF(NSKEW.EQ.0) GO TO 140
      N5A = N5+5*NSKEW
      N5B = N5A+2*NSKEW
  140 CONTINUE
C
C     CALL SEGLD(ISEG,IR1)
      CALL DPSM1
C
C     CALL SEGLD(MSEG,IRA)
C
      CALL DPSMA
C
C     CALL SEGLD(JSEG,IR2)
      CALL DPSM2
C
C     CALL SEGLD(KSEG,IR3)
      CALL DPSM3
C
C
C     CLOSE(5)
C     CLOSE(6)
C     IF(NDC.NE.0.AND.NSG.EQ.0) CLOSE(7)
C     IF(NTHS.EQ.1.OR.NTHS.EQ.3) CLOSE(40)
C     IF(NTHS.EQ.2.OR.NTHS.EQ.3) CLOSE(30)
C     CLOSE(3)
C     CLOSE(4)
C     CLOSE(10)
C     CLOSE(11)
  800 STOP
      END
C     BLOCK DATA
C     INTEGER*4 NA
C     COMMON /INTCN/ AA(19)
C     COMMON /NPARM/ NA(40)
C     COMMON /EDT/ ANF(58)
C
C     END
```

```
C FTN7X,L
C $EMA /ZAREA/,/OUTPS/,/OUTHS/
C     PROGRAM DREAD(5)
      SUBROUTINE DREAD
      COMMON /ZAREA/ Z(1500)
      COMMON /LPARM/ L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,MAXIP,MXIP,MXSH,
     *   LBBB,LDDD,NELS
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      COMMON /GCALP/ NCYCL,DT,NUPDAT,ITER
      COMMON /MMBDP/ NUMNP,NUMEL,NUMMAT,NVIS,NSKEW,NEQ,MBAND
      COMMON /ACCLF/ MTYPE,NUMAP,DTXA
      COMMON /PRELF/ NUMLP,NUMLH,NUMTP,NTYPE,DTXX
      COMMON /VELLF/ NUMVEL,NUMVH,NUMVTP,NVTYPE,DTXV
      COMMON /OUTPS/ NPFL,NDC,NSG,NPRINT,NPEL,NPMT,NPRT(1500),NPM(1500)
      COMMON /OUTHS/ NTHS,NHPEL,NHPMT,NHPRT(100),NHPM(100)
      COMMON /INTCN/ AA(15),TETA,BETA,GAMA,ALPA
      COMMON /BATCH/ IBATCH
      INTEGER LTITLE(40),LSUBTL(40),LTIME(15),IDATE(2),INAME(2),
     .       IPROB(10)
C     OPEN(5,FILE='NWOIP',STATUS='OLD')
C     OPEN(6,FILE='NWOOP::-19:4:200',STATUS='NEW')
C     OPEN(3,STATUS='SCRATCH')
C     OPEN(4,STATUS='SCRATCH')
C     OPEN(10,STATUS='SCRATCH')
C     OPEN(11,STATUS='SCRATCH')
      CALL DATE(MIN)
      CALL TIME(MIN2)
      READ(5,10) IBATCH
   10 FORMAT(I5)
      IF(IBATCH.EQ.1) GO TO 5400
      PRINT 4000
 4000 FORMAT(1X,"DATA TO APPEAR ON OUTPUT: ")
      PRINT 4100
 4100 FORMAT(/////,"INPUT USER'S NAME.")
      READ 4110,INAME
 4110 FORMAT(2A10)
      PRINT 4300
 4300 FORMAT(/"INPUT PROBLEM DESCRIPTION: ")
      READ 4310,IPROB
 4310 FORMAT(10A10)
      PRINT 4400
 4400 FORMAT(30(/))
      PRINT 5000
 5000 FORMAT(
     ."   ##########    #######    ########       ########     ####### "/
     ."   ##########    ########   #########     ##########    ########" /
     ."       ##        ##    ##   ##       ##   ##      ##    ##    ##" /
     ."       ##        ##    ##   ##       ##   ##      ##    ##    ##" /
     ."       ##        ########   ##       ##   ##########    ########" /
     ."       ##        #######    ##       #*   ##########    ####### " /
     ."       ##       ##         ##      ##   ##      ##   ##      " /
     ."       ##       ##         ##      ##   ##      ##   ##      " /
     ."       ##       ##         #########    ##      ##   ##      " /
     ."       ##       ##         ########     ##      ##   ##   "///)
C
      PRINT 5100,INAME
 5100 FORMAT(28X,"NAME : ",2A10)
      PRINT 5200,MIN
 5200 FORMAT(28X,"YEAR/MONTH/DAY : ",A10)
      PRINT 5250,MIN2
 5250 FORMAT(23X,"TIME : ",A10,"   (MOUNTAIN)")
      PRINT 5300,IPROB
 5300 FORMAT("JOB DESCRIPTION : ",10A10/)
C
C
C
 5400 WRITE(6,6100)
 6100 FORMAT(23(/))
      WRITE(6,6000)
 6000 FORMAT(
     .36X,"##########   #######    ########      ########    ####### "/
     .36X,"##########   ########   #########    ##########   ########"/
     .36X,"    ##       ##    ##   ##      ##   ##      ##   ##    ##"/
     .36X,"    ##       ##    ##   ##      ##   ##      ##   ##    ##"/
     .36X,"    ##       ########   ##      ##   ##########   ########"/
     .36X,"    ##       #######    ##      ##   ##########   ####### "/
     .36X,"    ##       ##         ##      ##   ##      ##   ##      "/
     .36X,"    ##       ##         ##      ##   ##      ##   ##      "/
     .36X,"    ##       ##         #########    ##      ##   ##      "/
     .36X,"    ##       ##         ########     ##      ##   ##   "///)
C      READ MAIN AND SUB TITLES
C
      READ(5,1005) LTITLE
      READ(5,1005) LSUBTL
C
      IF(IBATCH.EQ.1) GO TO 6150
      WRITE(6,6125) INAME
 6125 FORMAT(57X,"NAME : "2A10)
 6150 WRITE(6,6200) MIN
 6200 FORMAT(//52X,"YEAR/MONTH/DAY : ",A10)
      WRITE(6,6210) MIN2
 6210 FORMAT(57X,"TIME : ",A10,"   (MOUNTAIN)")
      IF(IBATCH.EQ.1) GO TO 6250
      WRITE(6,6225) IPROB
 6225 FORMAT(32X,"JOB DESCRIPTION : ",10A10/)
      GO TO 6350
 6250 WRITE(6,6300) LTITLE
 6300 FORMAT(//32X,"JOB DESCRIPTION : ",40A2)
 6350 WRITE(6,6100)
C
C
```

```
      WRITE(6,2005) LTITLE
      WRITE(6,2005) LSUBTL
C
C     RECORD START TIME
C
C     CALL FTIME(LTIME)
C     WRITE(6,1005) (LTIME(I),I=1,15)
C     WRITE(1,1005) (LTIME(I),I=1,15)
C
C     GENERAL OPTIONS
C
      READ (5,1010) NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
C
      WRITE(6,2010) NF,NTCSF,ISFG,IP
      WRITE(6,2015) NLNR,ICONST,NFG
C
C     GLOBAL CALCULATION PARAMETERS
C
      READ (5,1020) NCYCL,DT,NUPDAT,ITER
C
      WRITE(6,2020) NCYCL,DT,NUPDAT,ITER
C
C     MESH, MATERIAL AND BOUNDARY PARAMETERS
C
      READ (5,1030) NUMNP,NUMEL,NUMMAT,NVIS,NSKEW
C
      WRITE(6,2030) NUMNP,NUMEL,NUMMAT,NVIS,NSKEW
C
      IF(NFG.NE.1) GO TO 100
C
C     ACCELERATION LOADING FUNCTIONS ( NFG = 1 )
C
      READ (5,1040) MTYPE,NUMAP,DTXA
C
      WRITE(6,2040) MTYPE,NUMAP,DTXA
C
  100 IF(NFG.EQ.1.OR.NFG.EQ.3) GO TO 200
C
C     PRESSURE LOADING FUNCTIONS ( NFG = 2 OR 4 )
C
      READ (5,1050) NUMLP,NUMLH,NUMTP,NTYPE,DTXX
C
      WRITE(6,2050) NUMLP,NUMLH,NUMTP,NTYPE,DTXX
C
  200 IF(NFG.LT.3) GO TO 300
C
C     VELOCITY LOADING FUNCTIONS ( NFG = 3 OR 4 )
C
      READ (5,1060) NUMVEL,NUMVH,NUMVTP,NVTYPE,DTXV
C
      WRITE(6,2060) NUMVEL,NUMVH,NUMVTP,NVTYPE,DTXV
```

```
C
  300 CONTINUE
C
C     PARAMETERS REQUIRED FOR STORAGE USE
C
      READ (5,1070) IVMDK,IELCEN,MAXIP,MXSH
C
      WRITE(6,2070) IVMDK,IELCEN,MAXIP,MXSH
C
      NELS = 0
      IF(IVMDK.EQ.1) NELS = NUMEL
      IF(IELCEN.EQ.0) GO TO 400
      MAXIP = MAXIP+1
      MXIP = 1
      GO TO 450
  400 CONTINUE
      MXIP = MAXIP
  450 CONTINUE
C
      L1 = 1
      L2 = L1+1
      L3 = L2+2*8*MAXIP
      L4 = L3+8*MAXIP
      L5 = L4+MAXIP
      L6 = L5+MAXIP
      L7 = L6+MAXIP
      LBBB = L7-L1
C
      L8 = L7+16*MAXIP
C
      L9 = L8+MXSH*MXIP
      L10 = L9+MXIP
      LDDD = L10-L8
C
      LMAX = 1500
      IF(L10.LE.LMAX) GO TO 480
C
      WRITE(6,2075) LMAX,L10
      STOP
C
  480 CONTINUE
C
C     OUTPUT STRESS AND/OR MOTION PROFILES SPECIFICATIONS
C
      READ (5,1080) NPFL,NDC,NSG,NPRINT,NPEL,NPMT
C
      WRITE(6,2080) NPFL,NDC,NSG,NPRINT,NPEL,NPMT
C
C     IF(NDC.NE.0.AND.NSG.EQ.0) OPEN(7,FILE='DISTR::-17',STATUS='NEW')
C
      IF(NPFL.EQ.1.OR.NPFL.EQ.3) GO TO 8
```

```
C
      IF(NPFL.EQ.2) GO TO 17
C
      DO 6 I=1,NUMEL
      NPRT(I) = I
    6 CONTINUE
      IF(NPFL.EQ.4) GO TO 9
C
   17 CONTINUE
C
      DO 7 I=1,NUMNP
      NPM(I) = I
    7 CONTINUE
C
      GO TO 9
C
    8 CONTINUE
C
      READ (5,*)   (NPRT(I),I=1,NPEL)
C
      WRITE(6,2082)
      WRITE(6,2084) (NPRT(I),I=1,NPEL)
C
      IF(NPFL.EQ.3) GO TO 9
C
      READ(5,*)     (NPM(I),I=1,NPMT)
C
      WRITE(6,2086)
      WRITE(6,2088) (NPM(I),I=1,NPMT)
C
    9 CONTINUE
C
C     OUTPUT STRESS AND/OR MOTION TIME HISTORY SPECIFICATIONS
C
      READ (5,1090) NTHS,NHPEL,NHPMT
C
      WRITE(6,2090) NTHS,NHPEL,NHPMT
C
      IF(NTHS.EQ.0) GO TO 700
      IF(NTHS.EQ.1) GO TO 600
C
C     LIST SPECIFIED ELEMENT NUMBERS IN SEQUENTIAL ORDER
C       ( NTHS = 2 OR 3 )
C
C     OPEN(30,FILE='THSTR::-19:4:200',STATUS='NEW')
      WRITE(6,2092) NHPEL
      READ(5,*) (NHPRT(I),I=1,NHPEL)
      WRITE(6,2093)
      WRITE(6,2094) (NHPRT(I),I=1,NHPEL)
  600 CONTINUE
      IF(NTHS.EQ.2) GO TO 700
```

```
C
C     LIST SPECIFIED NODE NUMBERS IN SEQUENTIAL ORDER
C       ( NTHS = 1 OR 3 )
C
C     OPEN(40,FILE='THDIS::-19:4:200',STATUS='NEW')
      WRITE(6,2095) NHPMT
      READ(5,*) (NHPM(I),I=1,NHPMT)
      WRITE(6,2096)
      WRITE(6,2097) (NHPM(I),I=1,NHPMT)
C
  700 CONTINUE
C
C     NUMERICAL TIME INTEGRATION OPTIONS
C
      READ (5,1100) TETA,BETA,GAMA,ALPA
      WRITE(6,2100) TETA,BETA,GAMA,ALPA
C
 1005 FORMAT(40A2)
 1010 FORMAT(7I5)
 1020 FORMAT(I5,F10.0,2I5)
 1030 FORMAT(5I5)
 1040 FORMAT(2I5,F10.0)
 1050 FORMAT(4I5,F10.0)
 1060 FORMAT(4I5,F10.0)
 1070 FORMAT(4I5)
 1080 FORMAT(6I5)
 1090 FORMAT(3I5)
 1100 FORMAT(4F10.0)
C
 2005 FORMAT(40A2)
C
 2010 FORMAT(/// *   GENERAL OPTIONS                   *         //
     .*  .EQ.1   STATIC ANALYSIS                       *          /
     .*  .EQ.2   CONSOLIDATION ANALYSIS                *          /
     .*  .EQ.3   TWO-PHASE DYNAMIC ANALYSIS    ----- = *,I5      //
     .*  .EQ.0   TWO-PHASE DYNAMIC ANALYSIS            *          /
     .*  .EQ.1   CONSOLIDATION ANALYSIS                *          /
     .*  .EQ.2   ONE-PHASE SOLID DYNAMIC ANALYSIS      *          /
     .*  .EQ.3   ONE-PHASE FLUID DYNAMIC ANALYSIS    = *,I5      //
     .*  .EQ.0   NO INITIAL STRESS                     *          /
     .*  .EQ.1   SOLID AND FLUID INITIAL STRESS        *          /
     .*  .EQ.2   ONLY SOLID INITIAL STRESS             *          /
     .*  .EQ.3   ONLY FLUID INITIAL STRESS             *          /
     .*  .EQ.4   IMPOSED FLUID INITIAL STRESS -----  = *,I5      //
     .*  .EQ.1   PLANE STRESS ANALYSIS                 *          /
     .*  .EQ.2   PLANE STRAIN ANALYSIS                 *          /
     .*  .EQ.3   AXISYMMETRIC ANALYSIS                 *          /
     .*  .EQ.4   SPHERICAL SYMMETRY --------------- = *,I5      //)
 2015 FORMAT(
     .*  .EQ.0   LINEAR ELASTIC MATERIAL MODEL         *          /
     .*  .EQ.1   DECOUPLED ELASTO-PLASTIC MODEL        *          /
     .*  .EQ.2   CAP MODEL (AVAILABLE SOON)                  *          /
     .*  .EQ.3   AFWL ENGINEERING MODEL                      *          /
     .*  .EQ.4   UNIAXIAL STRAIN MODEL                       *          /
     .*  .EQ.5   ARA2D MODEL ----------------------- = *,I5            //
     .*  .EQ.0   LUMPED MASS                                 *          /
     .*  .EQ.1   CONSISTENT MASS  ----------------- = *,I5             //
     .*  .EQ.1   BASE ACCELERATIONS                          *          /
     .*  .EQ.2   PRESSURE LOAD                               *          /
     .*  .EQ.3   SPECIFIED VELOCITY                          *          /
     .*  .EQ.4   BOTH PRESSURE AND VELOCITY ------- = *,I5             //)
C
 2020 FORMAT(
     .*  NUMBER OF CYCLES -------------------------- = *,I5            //
     .*  GLOBAL TIME STEP -------------------------- = *,E11.4         //
     .*  NUMBER OF CYCLES BETWEEN UPDATES --------- = *,I5             //
     .*  NUMBER OF ITERATIONS --------------------- = *,I5            ///)
C
 2030 FORMAT(
     .*  NUMBER OF NODES --------------------------- = *,I5            //
     .*  NUMBER OF ELEMENTS ------------------------ = *,I5            //
     .*  NUMBER OF DIFFERENT MATERIALS ------------ = *,I5             //
     .*  NUMBER OF NODES ON TRANSMITTING BOUNDARY - = *,I5             //
     .*  NUMBER OF ELEMENT SIDES ON SKEW BOUNDARY - = *,I5            ///)
C
 2040 FORMAT(
     .*  ACCELERATION LOADING FUNCTIONS                  *             //
     .*  .EQ.0   CONSTANT TIME INCREMENTS                *              /
     .*  .EQ.1   SPECIFIED TIMES ------------------ = *,I5             //
     .*  NUMBER OF ACCEL. TIME PAIRS -------------- = *,I5             //
     .*  CONSTANT TIME INCREMENT ------------------ = *,E11.4         ///)
C
 2050 FORMAT(
     .*  PRESSURE LOADING FUNCTIONS                      *             //
     .*  TOTAL NUMBER OF LOAD POINTS -------------- = *,I5             //
     .*  NUMBER OF PRESSURE TIME HISTORIES -------- = *,I5             //
     .*  NUMBER OF PRESSURE-TIME PAIRS ------------ = *,I5             //
     .*  .EQ.0   CONSTANT TIME INCREMENTS                *              /
     .*  .EQ.1   SPECIFIED TIMES ------------------ = *,I5             //
     .*  CONSTANT TIME INCREMENT ------------------ = *,E11.4         ///)
C
 2060 FORMAT(
     .*  VELOCITY LOADING FUNCTIONS                      *             //
     .*  TOTAL NUMBER OF VELOCITY POINTS ---------- = *,I5             //
     .*  NUMBER OF VELOCITY TIME HISTORIES -------- = *,I5             //
     .*  NUMBER OF VELOCITY-TIME PAIRS ------------ = *,I5             //
     .*  .EQ.0   CONSTANT TIME INCREMENT                 *              /
     .*  .EQ.1   SPECIFIED TIMES ------------------ = *,I5             //
     .*  CONSTANT TIME INCREMENT ------------------ = *,E11.4         ///)
C
 2070 FORMAT(
     .*  PARAMETERS REQUIRED FOR STORAGE USE             *             //
     .*  .EQ.0   USE DISK AS SEQUENTIAL TAPE                *          /
     .*  .EQ.1   USE DISK AS VIRTUAL MEMORY ------- = *,I5       //
     .*  .EQ.0   COMPUTE STRESS AT INTEGRATION POINT. *          /
     .*  .EQ.1   COMPUTE STRESS AT ELEMENT CENTER - = *,I5       //
     .*  MAXIMUM NUMBER OF INTEGRATION POINTS ----- = *,I5       //
     .*  MAX. NUMBER OF STRESS/STRAIN HISTORY DATA  = *,I5      ///)
C
 2075 FORMAT(///
     .*  Z-DIMENSION SET IN PROGRAM --------------- = *,I10      //
     .*  REQUIRED Z-DIMENSION FOR PROBLEM --------- = *,I10     ///)
C
 2080 FORMAT(
     .*  OUTPUT PROFILE SPECIFICATIONS              *            //
     .*  .EQ.0   AT ALL NODES AND ELEMENTS          *             /
     .*  .EQ.1   AT SPECIFIED NODES AND ELEMENTS    *             /
     .*  .EQ.2   MOTIONS AT ALL NODES               *             /
     .*  .EQ.3   STRESSES AT SPECIFIED ELEMENTS     *             /
     .*  .EQ.4   STRESSES AT ALL ELEMENTS --------- = *,I5       //
     .*  .EQ.0   WRITE OUTPUT ON HARD DISK          *             /
     .*  .EQ.1   WRITE OUTPUT ON FLOPPY DISK ------ = *,I5       //
     .*  .EQ.0   WRITE OUTPUT IN ONE FILE           *             /
     .*  .EQ.1   WRITE OUTPUT IN SPECIFIED FILES -- = *,I5       //
     .*  NUMBER OF CYCLES BETWEEN EACH OUTPUT ----- = *,I5       //
     .*  NUMBER OF ELEMENTS IN OUTPUT STRESS PROF.  = *,I5       //
     .*  NUMBER OF NODES IN OUTPUT MOTION PROFILES  = *,I5      ///)
C
 2082 FORMAT(/ *  ELEMENTS TO BE PRINTED  ----------- *           /)
 2084 FORMAT(15I5)
 2086 FORMAT(/ *  NODAL POINTS TO BE PRINTED  ------- *           /)
 2088 FORMAT(15I5)
C
 2090 FORMAT(
     .*  OUTPUT TIME HISTORY SPECIFICATIONS         *            //
     .*  .EQ.0   DO NOT PRINT TIME HISTORY DATA     *             /
     .*  .EQ.1   MOTION TIME HISTORIES              *             /
     .*  .EQ.2   STRESS TIME HISTORIES              *             /
     .*  .EQ.3   MOTION AND STRESS TIME HISTORIES - = *,I5       //
     .*  NUMBER OF ELEMENTS TO BE PRINTED --------- = *,I5       //
     .*  NUMBER OF NODES TO BE PRINTED ------------ = *,I5      ///)
C
 2092 FORMAT(/
     .*  NUMBER OF ELEMENTS TO BE PRINTED --------- = *,I5        /)
 2093 FORMAT(/ *  ELEMENTS TO BE PRINTED  ----------- *           /)
 2094 FORMAT(15I5)
 2095 FORMAT(/
     .*  NUMBER OF NODES TO BE PRINTED ------------ = *,I5        /)
 2096 FORMAT(/ *  NODAL POINTS TO BE PRINTED  ------- *           /)
 2097 FORMAT(15I5)
C
 2100 FORMAT(/// *   INTEGRATION CONSTANTS          *            //
     .*  TETA ------------------------------------- = *,E10.3    //
     .*  BETA  ------------------------------------- = *,E10.3   //
     .*  GAMA  ------------------------------------- = *,E10.3   //
     .*  ALPA  ------------------------------------- = *,F10.3  ///)
C
C
C
C
C     CALL SEGRT
      RETURN
      END
```

```
C FTN7X,L
C C $EMA /WAREA/
C      PROGRAM DPSM1(5)
       SUBROUTINE DPSM1
C      INTEGER*4 N1,N2,N3,N4,N5,NN,N5A,N5B
C      INTEGER LTIME(15)
       COMMON /WAREA/ A(60000)
       COMMON /NPARM/ N1,N2,N3,N4,N5,NN(33),N5A,N5B
       COMMON /GCALP/ NCYCL,DT,NUPDAT,ITER
       COMMON /MMBDP/ NUMNP,NUMEL,NUMMAT,NVIS,NSKEW,NEQ,MBAND
C
C      WRITE(1,*) *BEFORE NODE*
       CALL NODE(A(N1),A(N2),A(N3),NUMNP,NEQ)
C      WRITE(1,*) *AFTER NODE*
C
       CALL EDATA(A(N1),NUMEL,MBAND,NUMNP,A(N4),NUMMAT,A(N5),A(N5A),
      *   NSKEW,A(N2),A(N3))
C      WRITE(1,*) *AFTER EDATA*
C
       CALL CNSTS(DT)
C      CALL FTIME(LTIME)
C      WRITE(1,1) (LTIME(I),I=1,15)
C    1 FORMAT(*AFTER DPSM1---*,15A2)
C
C      CALL SEGRT
       RETURN
       END
       SUBROUTINE NODE(ID,R,Z,NUMNP,NEQ)
C
C      NODAL POINT INPUT AND GENERATION
C
       DIMENSION R(1),Z(1),ID(NUMNP,4),NID(4)
C      EMA R,Z,ID
C
       ND = 3
       KO = 1
C
       DO 5 I=1,4
     5 NID(I) = 0
C
    11 READ (5,100) N,(ID(N,I),I=1,4),R(N),Z(N),KN
       DO 60 I=1,4
       IF (ID(N,I)) 61,62,63
    61 NID(I) = -1
       ID(N,I) = 1
       GO TO 60
    62 IF(NID(I).EQ.-1) ID(N,I) = 1
       GO TO 60
    63 NID(I) = 0
    60 CONTINUE
```

```
      IF(KO.EQ.1) GO TO 12
C
C       CHECK IF GENERATION NEEDED
C
      IF (KN) 10,10,20
   12 KO = 0
   10 CONTINUE
      NUMINT = 1
      GO TO 15
C
C       GENERATE NEW NODES
C
   20 NUMINT = (N-NI)/KN
      DR = (R(N)-R(NI))/NUMINT
      DZ = (Z(N)-Z(NI))/NUMINT
      NUMINT = NUMINT-1
      DO 21 J=1,NUMINT
      NN = NI+J*KN
      R(NN) = R(NN-KN)+DR
      Z(NN) = Z(NN-KN)+DZ
C
C       SET BOUNDARY CODES...SAME AS FIRST JOINT SERIES
C
      DO 22 JJ=1,4
      IF(ID(NI,JJ)-1) 24,26,25
C
C       GENERATE NEW MASTER NODES
C
   25 ID(NN,JJ) = ID(NI,JJ)+J*KN
      GO TO 22
   26 ID(NN,JJ) = ID(NI,JJ)
      GO TO 22
   24 ID(NN,JJ) = 0
   22 CONTINUE
   21 CONTINUE
   15 NI = N
C
C       CHECK FOR LAST NODAL POINT
C
      IF(NUMNP-NI) 13,13,11
   13 CONTINUE
C
C       PRINT ALL NODAL DATA
C
      WRITE(6,202)
      WRITE(6,204)
      DO 50 N=1,NUMNP
   50 WRITE(6,203) N,(ID(N,I),I=1,4),R(N),Z(N)
C
C       SPECIFY GLOBAL EQUATION NUMBER FOR FREE DOFS
C
```

```
      NEQ = 0
      NEQH = 0
      DO 75 I=1,NUMNP
      DO 70 J=1,4
      IF(ID(I,J)) 68,65,68
   65 NEQ = NEQ+1
      ID(I,J) = NEQ
      GO TO 70
   68 ID(I,J) = -1
   70 CONTINUE
   75 CONTINUE
      NEQH = NEQ
      WRITE(6,205) NEQ,NEQH
C
      RETURN
C
  100 FORMAT(5I5,2F10.0,I5)
  104 FORMAT(I5,4F10.4)
  202 FORMAT(26H1COMPLETE NODAL POINT DATA  //  )
  203 FORMAT(I5,4I5,2F10.3,I5)
  204 FORMAT(
     .*  NODE   BOUNDARY CODES       NODAL COORDINATES */
     .*   NO   ISX  ISY  IFX  IFY       XA         YA    */)
  205 FORMAT(//30H NUMBER OF EQUATIONS .......       ,I10//
     *         30H NUMBER OF ROWS OF H .......       ,I10//)
C
      END
C $EMA /MDCPL/
      SUBROUTINE EDATA(ID,NUMEL,MBAND,NUMNP,PROP,NUMMAT,ISKEW,ACS,
     *   NSKEW,X,Y)
      COMMON /EDT/ NEL,IXS(16),IXF(16),NP(8),MIDSID(4),INN(3),
     *   ISF(8),MC,MCC
      COMMON /MDCPL/ BLN(11,2,5),BUN(11,2,5),TYF(11,3,5),NLP(5),
     *   NUP(5),NTP(5),NCO(5)
      DIMENSION ID(NUMNP,4),AA(58),LM(32),PROP(30,1),ISKEW(5,1),
     *   ACS(2,1),X(1),Y(1)
C     EMA ID,PROP,ISKEW,ACS,X,Y
      EQUIVALENCE (NEL,AA(1))
C
      REWIND 3
      MBAND = 0
      MBANDH = 0
      NEL = 0
      CALL IZERO(ISF,8)
      CALL IZERO(INN,3)
C
C
      WRITE(6,2000)
 2000 FORMAT(
     .*   NEL  MAT  KS   KF  INTR INTS   I    J    K    L*,
     .*   M1   M2   M3   M4*/)
```

```
C     WRITE(6,*) "BEFORE DO 650"
      DO 650 II=1,NUMEL
      READ(5,1007) NEL,MAT,KS,KF,INTR,INTS,NP
      ISF(1) = MAT
      ISF(3) = KS
      ISF(4) = KF
      ISF(5) = INTR
      ISF(6) = INTS
C
      NC = 4
      DO 35 I=1,4
      IF(NP(I+4).EQ.0) GO TO 30
      NC = NC+1
      MIDSID(I) = 1
      NP(NC) = NP(I+4)
      GO TO 35
   30 MIDSID(I) = 0
   35 CONTINUE
      NCC = 2*NC
      MC = NC
      MCC = NCC
      WRITE(6,1007) NEL,MAT,KS,KF,INTR,INTS,NP
C
      CALL IZERO(LM,32)
      L = 0
      DO 620 I=1,NC
      N = NP(I)
      DO 610 J=1,4
      L = L+1
      LM(L) = ID(N,J)
  610 CONTINUE
  620 CONTINUE
C
      MM = 1
      DO 625 I=1,NC
      IXS(I) = LM(MM)
      IXS(I+NC) = LM(MM+1)
      IXF(I) = LM(MM+2)
      IXF(I+NC) = LM(MM+3)
      MM = MM+4
  625 CONTINUE
C
      WRITE (3) AA
C
      NEE = 4*NC
C
      DO 640 I=1,NEE
      L = LM(I)
      IF(L) 640,640,630
  630 DO 635 J=1,NEE
      N = LM(J)
      IF(N) 635,635,632
  632 KL = IABS(N-L)
      IF(KL.GT.MBAND) MBAND = KL
  635 CONTINUE
  640 CONTINUE
  650 CONTINUE
C     WRITE(6,*) "AFTER DO 650"
C
      MBAND = MBAND+1
      MBANDH = MBAND
      WRITE(6,2010) MBAND,MBANDH
C
C      READ SKEW BOUNDARY CONDITIONS
C
      IF(NSKEW.EQ.0) GO TO 680
      DO 660 I=1,NSKEW
      READ(5,1010) MEL,IND,JND,ISN,MSF
      WRITE(6,1010) MEL,IND,JND,ISN,MSF
      ISKEW(1,I) = MEL
      XI = X(IND)
      XJ = X(JND)
      YI = Y(IND)
      YJ = Y(JND)
      CALL BSKEW(ISN,XI,YI,XJ,YJ,INX,JNX,KNX,MIDSID,CALPA,SALPA)
      ISKEW(2,I) = INX
      ISKEW(3,I) = JNX
      ISKEW(4,I) = KNX
      ISKEW(5,I) = MSF
      ACS(1,I) = CALPA
      ACS(2,I) = SALPA
  660 CONTINUE
  680 CONTINUE
 1010 FORMAT(5I5)
C
C       READ MATERIAL PROPERTIES
C
      WRITE(6,2020)
      DO 700 I=1,NUMMAT
      READ(5,1020) (PROP(J,I),J=1,30)
      WRITE(6,2022) (PROP(J,I),J=1,8)
      WRITE(6,2024) (PROP(J,I),J=9,16)
      WRITE(6,2026) (PROP(J,I),J=17,24)
      WRITE(6,2028) (PROP(J,I),J=25,30)
  700 CONTINUE
 2020 FORMAT(
     .*MATERIAL PARAMETERS *//)
C
 2022 FORMAT(
     .*    ELS = *,F11.0  /
     .*   TENS = *,F11.0  /
     .* STIFAC = *,F11.2  /
     .* SHEFAC = *,F11.2  /
     .*    PMN = *,F11.2  /
     .*     BK = *,F11.1  /
     .*      G = *,F11.1  /
     .*    D8  = *,F11.2  //)
C
 2024 FORMAT(
     .*    D9  = *,E11.4  /
     .*    D10 = *,E11.4  /
     .*    D11 = *,E11.4  /
     .*    D12 = *,E11.4  /
     .*    D13 = *,E11.4  /
     .*    D14 = *,E11.4  /
     .*    D15 = *,E11.4  /
     .*    D16 = *,E11.4  //)
C
 2026 FORMAT(
     .*    D17 = *,E11.4  /
     .*    D18 = *,E11.4  /
     .*     RO = *,E11.4  /
     .*    ROF = *,E11.4  /
     .*    D21 = *,E11.4  /
     .*    D22 = *,E11.4  /
     .*    D23 = *,E11.4  /
     .*    D24 = *,E11.4  //)
C
 2028 FORMAT(
     .*    D25 = *,E11.4  /
     .*     CM = *,F11.1  /
     .*    POR = *,F11.3  /
     .*    RXX = *,E11.4  /
     .*    RYY = *,E11.4  /
     .*    RXY = *,E11.4  //)
C
C
      IF(NLNR.NE.1) GO TO 50
      READ(5,1) NUMAT
1     FORMAT(I5)
      DO 40 II=1,NUMAT
C
      READ(5,1) NLPC
      DO 10 I=1,NLPC
      READ(5,2) BLN(I,1,II),BLN(I,2,II)
10    CONTINUE
2     FORMAT(2F10.0)
C
      READ(5,1) NUPC
      DO 20 I=1,NUPC
      READ(5,2) BUN(I,1,II),BUN(I,2,II)
20    CONTINUE
C
```

```
      READ(5,1) NTPC
      DO 25 I=1,NTPC
      READ(5,3) TYF(I,1,II),TYF(I,2,II),TYF(I,3,II)
25    CONTINUE
3     FORMAT(3F10.0)
C
      NLP(II)=NLPC
      NUP(II)=NUPC
      NTP(II)=NTPC
C
40    CONTINUE
50    CONTINUE
C
      IF(NLNR.NE.4) GO TO 150
C
      READ(5,1) NUMAT
      DO 140 II=1,NUMAT
C
      READ(5,1) NLPC
      DO 110 I=1,NLPC
      READ(5,2) BLN(I,1,II),BLN(I,2,II)
110   CONTINUE
C
      READ(5,1) NUPC
      IF(NUPC.EQ.0) GO TO 130
      DO 120 I=1,NUPC
      READ(5,2) BUN(I,1,II),BUN(I,2,II)
120   CONTINUE
C
130   CONTINUE
      NTP(II)=1
      NUP(II)=NUPC
      IF(NUPC.EQ.0) NUP(II)=1
      NLP(II)=NLPC
C
140   CONTINUE
C
150   CONTINUE
C
 1020 FORMAT(8E10.3)
      RETURN
C
 1007 FORMAT(14I5)
 2010 FORMAT(//*..... BAND WIDTH ............. *,I10/
     *         *..... BAND WIDTH FOR  H ...... *,I10)
      END
      SUBROUTINE CNSTS(DT)
      COMMON /INTCN/ A(15),TETA,BETA,GAMA,ALPA
      TETE = TETA*TETA
      TAU = TETA*DT
      TATA = TAU*TAU
      A(1) = 1.0/(BETA*TATA)
      A(2) = TAU
      A(3) = (0.5-BETA)*TATA
      A(4) = GAMA/(BETA*TAU)
      A(5) = (1.0-BETA/GAMA)*TAU
      A(6) = (0.5-BETA/GAMA)*TATA
      A(7) = 1.0/(TETE*TETA)
      A(8) = (1.0-1.0/TETE)*DT
      A(9) = 0.5*DT*DT*(1.0-1.0/TETA)
      A(10) = GAMA/(BETA*TETE*TAU)
      A(11) = 1.0-GAMA/(BETA*TETE)
      A(12) = (1.0-GAMA/(2.0*BETA*TETA))*DT
      A(13) = 1.0/(BETA*TETA*TATA)
      A(14) = -1.0/(BETA*TETA*TAU)
      A(15) = 1.0-1.0/(2.0*BETA*TETA)
      RETURN
      END
      SUBROUTINE BSKEW(ISN,XI,YI,XJ,YJ,INX,JNX,KNX,MIDSID,CALPA,SALPA)
      DIMENSION MIDSID(4)
      GO TO (10,20,30,40) ISN
   10 INX = 2
      JNX = 1
      CALL ASKNX(KNX,MIDSID,ISN)
      GO TO 50
   20 INX = 3
      JNX = 2
      CALL ASKNX(KNX,MIDSID,ISN)
      GO TO 50
   30 INX = 3
      JNX = 4
      CALL ASKNX(KNX,MIDSID,ISN)
      GO TO 50
   40 INX = 4
      JNX = 1
      CALL ASKNX(KNX,MIDSID,ISN)
   50 CONTINUE
      XL = (XJ-XI)**2+(YJ-YI)**2
      XL = SQRT(XL)
      CALPA = (XJ-XI)/XL
      SALPA = (YJ-YI)/XL
      RETURN
      END
      SUBROUTINE ASKNX(KNX,MIDSID,ISN)
      DIMENSION MIDSID(4)
      KNX = 0
      IF(MIDSID(ISN).EQ.0) RETURN
      DO 100 I=1,ISN
      KNX = KNX+MIDSID(I)
  100 CONTINUE
      IF(KNX.NE.0) KNX = KNX+4
      RETURN
```

```
C FTN7X,L
C $EMA /OUTPS/,/OUTHS/
C     PROGRAM DPSMA(5)
      SUBROUTINE DPSMA
C     INTEGER*4 N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15,
C    *  N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,N29,N30
C     INTEGER*4 N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B,NLAST
      COMMON /NPARM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,
     * N14,N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,
     * N29,N30,N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B
      COMMON /LPARM/ L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,MAXIP,MXIP,MXSH,
     *   LBBB,LDDD,NELS
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      COMMON /GCALP/ NCYCL,DT,NUPDAT,ITER
      COMMON /MMBDP/ NUMNP,NUMEL,NUMMAT,NVIS,NSKEW,NEQ,MBAND
      COMMON /ACCLF/ MTYPE,NUMAP,DTXA
      COMMON /PRELF/ NUMLP,NUMLH,NUMTP,NTYPE,DTXX
      COMMON /VELLF/ NUMVEL,NUMVH,NUMVTP,NVTYPE,DTXV
      COMMON /OUTPS/ NPFL,NDC,NSG,NPRINT,NPEL,NPMT,NPRT(1500),NPM(1500)
      COMMON /OUTHS/ NTHS,NHPEL,NHPMT,NHPRT(100),NHPM(100)
C     COMMON /NALLO/ NSDFL(8,9,2,2)
      TOTM = 60000.
C
C     IMASS = 0  DO NOT SUPPRESS MASS
C             1          SUPPRESS MASS
C
      IMASS = 0
C
      N6 = N5B+NEQ*MBAND
      N7 = N6+NEQ*MBAND
      N8 = N7
      IF(NTCSF.NE.2) N8 = N7+NEQ*MBAND
      N9 = N8+NEQ*MBAND
      IF(ICONST.EQ.0) N9 = N8+NEQ
      IF(NTCSF.EQ.1.OR.IMASS.EQ.1) N9 = N8
      N10 = N9+NEQ
      N11 = N10
      IF(ICONST.EQ.1.AND.IMASS.EQ.0.AND.NTCSF.NE.1)
     *  N11 = N10+2*NEQ
      N12 = N11
      IF(NTCSF.NE.2) N12 = N11+2*NEQ
      N13 = N12+NEQ
      N14 = N13+NEQ
      N15 = N14+NEQ
      N16 = N15+NEQ
      N17 = N16+NEQ
      N18 = N17+NEQ
      N19 = N18+NEQ
      N20 = N19+NEQ
C
```

```
      IF(NFG.EQ.1) GO TO 180
      IF(NFG.EQ.3) GO TO 150
      N21 = N20
      N22 = N20
C
      N23 = N22+NUMLH*NUMTP
      N24 = N23+3*NUMLP
      N25 = N24+NUMLP
      N26 = N25+NUMTP
      N27 = N26+NEQ
C
      GO TO 200
C
  150 CONTINUE
      N21 = N20
      N22 = N20
      N23 = N20
      N24 = N20
      N25 = N20
      N26 = N20
      N27 = N20
      GO TO 200
C
  180 CONTINUE
      N21 = N20+NUMAP
      IF(MTYPE.EQ.0) N21 = N20
      N22 = N21+NUMAP
C
      N23 = N22
      N24 = N22
      N25 = N22
      N26 = N22
      N27 = N22
C
  200 CONTINUE
C
      N28 = N27
      IF(ISFG.EQ.1.OR.ISFG.EQ.2) N28 = N27+4*MXIP*NUMEL
      N29 = N28
      IF(ISFG.EQ.1.OR.ISFG.GE.3) N29 = N28+MXIP*NUMEL
      N30 = N29+58*NUMEL
      N31 = N30+LBBB*NELS
      N32 = N31+LDDD*NELS*2
C
      N33 = N32
      N34 = N32
      IF(NVIS.EQ.0) GO TO 300
      N33 = N32+NVIS
      N34 = N33+NVIS
  300 CONTINUE
      N35 = N34
      N36 = N34
      N37 = N34
      N38 = N34
      N39 = N34
      IF(NUMVEL.EQ.0) GO TO 400
      N35 = N34+NUMVEL
      N36 = N35+NUMVEL
      N37 = N36+NUMVEL
      N38 = N37+NUMVTP
      N39 = N38+NUMVH*NUMVTP
  400 CONTINUE
      NLAST = N39
C
      WRITE(6,2095) TOTM,NLAST
C
      IF(NLAST.GT.TOTM) STOP
C
C
C
 2095 FORMAT(/ * INFORMATION ABOUT PROGRAM ARRAY DIMENSION  *    /
     .  .*   PROGRAM DIMENSION  ---------------------  = *,F10.0 /
     .   *   REQUIRED DIMENSION ---------------------  = *,I10  //)
C
C     CALL SEGRT
      RETURN
      END
```

```
C FTN7X,L
C $EMA /WAREA/,/ZAREA/
C     PROGRAM DPSM2(5)
      SUBROUTINE DPSM2
C     INTEGER*4 N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,
C    *  N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,N29
C     INTEGER*4 N30,N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B
C     INTEGER LTIME(15)
      COMMON /WAREA/ A(60000)
      COMMON /NPARM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,
     * N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,
     * N27,N28,N29,N30,N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B
      COMMON /INTCN/ B(19)
C     COMMON /GAUSQ/ ZETA(4,4),W(4,4)
      COMMON /MMBDP/ NUMNP,NUMEL,NUMMAT,NVIS,NSKEW,NEQ,MBAND
      COMMON /PRELF/ NUMLP,NUMLH,NUMTP,NTYPE,DTXX
      COMMON /VELLF/ NUMVEL,NUMVH,NUMVTP,NVTYPE,DTXV
      COMMON /ZAREA/ Z(1500)
      COMMON /LPARM/ L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,MAXIP,MXIP,MXSH,
     *   LBB,LDD,NELS
      CALL DATA
C
      C2 = B(1)
      C3 = B(4)
C     WRITE(6,*) *NPARM = *
C     WRITE(6,*) N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11
C     WRITE(6,*) N27,N28,N29,N30,N31,N32,N33,N34,N35,N36
C     WRITE(6,*) N37,N38,N5A,N5B
C
      CALL KSTAR(A(N2),A(N3),A(N4),A(N5B),A(N8),A(N7),A(N10),A(N11),
     * NEQ,C2,C3,A(N30),NUMNP,A(N1),A(N32),A(N33),A(N5),A(N5A),NUMEL,
     * NVIS,Z(L1),Z(L2),Z(L3),Z(L4),Z(L5),Z(L6),Z(L1),Z(L7),MAXIP,
     * MXIP,LBB,NELS,A(N27),A(N28),NUMLH,A(N20),A(N21),A(N22),A(N23),
     * A(N24),A(N25),A(N26),NUMVTP,A(N34),A(N35),A(N36),A(N37),A(N38))
C     CALL FTIME(LTIME)
C     WRITE(1,1) (LTIME(I),I=1,15)
C   1 FORMAT(*AFTER DPSM2---*,15A2)
C
C     CALL SEGRT
      RETURN
C
      END
C     BLOCK DATA
      SUBROUTINE DATA
      COMMON /GAUSQ/ ZETA(4,4),W(4,4)
      DATA ZETA/
     1 .000000000000, .000000000000, .000000000000, .000000000000,
     2-.5773502691896, .5773502691896, .000000000000, .000000000000,
     3-.7745966692415, .000000000000, .7745966692415, .000000000000,
     4-.8611363115941,-.3399810435849, .3399810435849, .8611363115941/
```

```
      DATA W/
     12.0000000000000, .0000000000000, .0000000000000, .0000000000000,
     21.0000000000000,1.0000000000000, .0000000000000, .0000000000000,
     3 .5555555555556, .8888888888889, .5555555555556, .0000000000000,
     4 .3478548451375, .6521451548625, .6521451548625, .3478548451375/
C
      RETURN
      END
      SUBROUTINE KSTAR(XA,YA,PROP,GKO,GM,GH,MBM,MBD,NEQ,C2,C3,DBBB,
     *   NUMNP,ID,IVIS,VISC,ISKEW,ACS,NUMEL,NVIS,NINT,B,EN,XBAR,YBAR,
     *   WIWJ,BB,XTT,MAXIP,MXIP,LBB,NELS,SRT,PRF,NUMLH,TS,GACL,DYL,
     *   IEQH,CINT,TD,UNV,NUMVTP,KVEL,KEQH,VINT,TV,SVEL)
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      COMMON /GCALP/ NCYCL,DT,NUPDAT,ITER
      COMMON /MMBDP/ NUMNX,NUMEX,NUMMAT,NVIX,NSKEX,NEQX,MBAND
      COMMON /PRELF/ NUMLP,NUMLX,NUMTP,NTYPE,DTXX
      COMMON /VELLF/ IVEL,NUMVH,NUMVTX,NVTYPE,DTXV
      COMMON /EDT/ NEL,IXS(16),IXF(16),NP(8),MIDSID(4),INN(3),
     *   ISF(8),NC,NCC
      DIMENSION GM(NEQ,1),GKO(NEQ,1),PROP(30,1),XA(1),YA(1),
     *   XX(8),YY(8),EMS(16,16),EMC(16,16),EJ(17),
     *   EMF(16,16),H(16,16),EE(16,16),C(16,16),E(16,16),
     *   GH(NEQ,1),MBM(1),MBD(1),D(4,4),EST(16,16),
     *   ID(NUMNP,1),IVIS(NVIS),VISC(NVIS),HEAD(20),
     *   ISKEW(5,1),ACS(2,1),B(2,8,MAXIP),EN(8,MAXIP),XBAR(MAXIP),
     *   YBAR(MAXIP),WIWJ(MAXIP),BB(LBB),XTT(16,MAXIP),AA(58),
     *   DBBB(LBB,NELS),SRT(4,MXIP,1),PRF(MXIP,1),TS(1),GACL(1),
     *   DYL(NUMLH,1),IEQH(3,1),CINT(1),TD(1),UNV(1),KVEL(1),KEQH(1),
     *   VINT(1),TV(1),SVEL(NUMVTP,1)
C     EMA XA,YA,PROP,GKO,GM,GH,MBM,MBD,DBBB,ID,IVIS,VISC,ISKEW,ACS,
C    *   NINT,B,EN,XBAR,YBAR,WIWJ,BB,XTT,SRT,PRF,TS,GACL,DYL,IEQH,
C    *   CINT,TD,UNV,KVEL,KEQH,VINT,TV,SVEL
      EQUIVALENCE (NEL,AA(1))
C
C
      REWIND 3
      IF(NELS.EQ.0)  REWIND 4
C
C     INITIALIZE
C
      IF(ICONST.EQ.0) GO TO 50
      CALL TZERO(GM,NEQ,MBAND)
      GO TO 60
   50 CALL EZERO(GM,NEQ)
   60 CALL TZERO(GKO,NEQ,MBAND)
      IF(NTCSF.NE.2)  CALL TZERO(GH,NEQ,MBAND)
C
      NSK = 1
      NSM = 1
      NSE = 1
C
```

```
      DO 500  NN = 1,NUMEL
C
C
      READ (3) AA
C
      MAT = ISF(1)
      KS = ISF(3)
      KF = ISF(4)
      INTR = ISF(5)
      INTS = ISF(6)
      BK = PROP(6,MAT)
      G = PROP(7,MAT)
C
C
      CM = PROP(26,MAT)
      RO = PROP(19,MAT)
      ROF = PROP(20,MAT)
      POR = PROP(27,MAT)
      RXX = PROP(28,MAT)
      RYY = PROP(29,MAT)
      RXY = PROP(30,MAT)
C
      DO 100 I=1,NC
      N = NP(I)
      XX(I) = XA(N)
      YY(I) = YA(N)
  100 CONTINUE
C
      NINT = INTR*INTS
C
      CALL TWOD(XX,YY,MIDSID,INTR,INTS,NEL,B,EN,EJ,XBAR,YBAR,NC,
     *   WIWJ,IP,0)
C
      IF(MXIP.NE.1) GO TO 105
      N = MAXIP
      NOIP = N-1
      CALL TWOD(XX,YY,MIDSID,1,1,NEL,B,EN,EJ,XBAR,
     *  YBAR,NC,WIWJ,IP,NOIP)
  105 CONTINUE
C
C     THIS OPTION IS GOOD FOR CYBER
C
      IF(NELS.EQ.0)  WRITE (4) BB
C
      IF(NELS.EQ.NUMEL)  CALL WREAD(DBBB(1,NN),LBB,BB,1)
C
C
C      COMPUTE ELEMENT ELASTIC STIFFNESS MATRIX AND ASSEMBLE
C
      IF(NLNR.NE.0) GO TO 130
      CALL ELAST(BK,G,D)
      CALL SZERO(EST,16*16)
C
      DO 110 I=1,NINT
      CALL ELSTF(B(1,1,I),D,EN(1,I),XBAR(I),WIWJ(I),NC,NCC,EST,IP,
     *   0 )
  110 CONTINUE
C
      IF(NSKEW.EQ.0) GO TO 120
      IF(NSK.GT.NSKEW) GO TO 120
      IF(NN.NE.ISKEW(1,NSK)) GO TO 120
      IF(ISKEW(5,NSK).EQ.1) GO TO 115
      INX = ISKEW(2,NSK)
      JNX = ISKEW(3,NSK)
      KNX = ISKEW(4,NSK)
      CSC = ACS(1,NSK)
      CSS = ACS(2,NSK)
      CALL MSKEW(EST,16,NC,NCC,INX,JNX,KNX,3,CSC,CSS)
  115 NSK = NSK+1
  120 CONTINUE
C
C     CALL ALPHA(EST)
C
      CALL ESMBL(GKO,DUM,EST,DUM,IXS,IXS,NEQ,MBAND,16,2,NCC,NCC)
C
C
      IF(NTCSF.EQ.1) GO TO 160
C
C      ASSEMBL GLOBAL MASS MATRIX
C
  130 CALL MASSM(EMS,EMC,EMF,EN,WIWJ,ICONST,RO,ROF,POR,NC,NCC,NINT,
     *   KS,KF,NTCSF)
C
      IF(NSKEW.EQ.0) GO TO 140
      IF(NSM.GT.NSKEW) GO TO 140
      IF(NN.NE.ISKEW(1,NSM)) GO TO 140
      INX = ISKEW(2,NSM)
      JNX = ISKEW(3,NSM)
      KNX = ISKEW(4,NSM)
      MSF = ISKEW(5,NSM)
      CSC = ACS(1,NSM)
      CSS = ACS(2,NSM)
      IF(MSF.NE.1)  CALL MSKEW(EMS,16,NC,NCC,INX,JNX,KNX,3,CSC,CSS)
      IF(KF.EQ.1.OR.NTCSF.EQ.2) GO TO 135
      CALL MSKEW(EMC,16,NC,NCC,INX,JNX,KNX,MSF,CSC,CSS)
      IF(MSF.NE.2)  CALL MSKEW(EMF,16,NC,NCC,INX,JNX,KNX,3,CSC,CSS)
  135 NSM = NSM+1
  140 CONTINUE
C
      IF(ICONST.EQ.0) GO TO 150
C
C      CONSISTENT MASS MATRIX
```

```
C
C
      CALL ESMBL(GM,DUM,EMS,DUM,IXS,IXS,NEQ,MBAND,16,2,NCC,NCC)
C
      IF(KF.EQ.1.OR.NTCSF.EQ.2) GO TO 500
      CALL ESMBL(GM,DUM,EMF,DUM,IXF,IXF,NEQ,MBAND,16,2,NCC,NCC)
C
      CALL ESMBL(GM,DUM,EMC,DUM,IXS,IXF,NEQ,MBAND,16,2,NCC,NCC)
C
      CALL ESMBL(GM,DUM,EMC,DUM,IXF,IXS,NEQ,MBAND,16,2,NCC,NCC)
C
      GO TO 160
C
C
C      LUMPED MASS MATRIX
C
  150 CALL ESMBL(DUM,GM,DUM,EMS,IXS,DUM,NEQ,DUM,16,3,NCC,DUM)
C
      IF(KF.EQ.1.OR.NTCSF.EQ.2) GO TO 500
      CALL ESMBL(DUM,GM,DUM,EMF,IXF,DUM,NEQ,DUM,16,3,NCC,DUM)
C
C      COMPUTE H, EE, E, AND C  MATRIX AND ASSEMBL IN GLOBAL
C
  160 CONTINUE
C
      CALL ELH(H,EN,WIWJ,NINT,RXX,RYY,RXY,NCC,NC)
C
      CALL ECTTX(B,EN,XBAR,IP,NC,NCC,XTT,EE,E,C,NINT,WIWJ,CM)
C
      IF(NSKEW.EQ.0) GO TO 180
      IF(NSE.GT.NSKEW) GO TO 180
      IF(NN.NE.ISKEW(1,NSE)) GO TO 180
      INX = ISKEW(2,NSE)
      JNX = ISKEW(3,NSE)
      KNX = ISKEW(4,NSE)
      MSF = ISKEW(5,NSE)
      CSC = ACS(1,NSE)
      CSS = ACS(2,NSE)
      IF(MSF.NE.1) CALL MSKEW(EE,16,NC,NCC,INX,JNX,KNX,3,CSC,CSS)
      CALL MSKEW(C,16,NC,NCC,INX,JNX,KNX,MSF,CSC,CSS)
      IF(MSF.EQ.2) GO TO 175
      CALL MSKEW(H,16,NC,NCC,INX,JNX,KNX,3,CSC,CSS)
      CALL MSKEW(E,16,NC,NCC,INX,JNX,KNX,3,CSC,CSS)
  175 NSE = NSE+1
  180 CONTINUE
C
      CALL ESMBL(GH,DUM,H,DUM,IXF,IXF,NEQ,MBAND,16,2,NCC,NCC)
C
      CALL ESMBL(GKO,DUM,EE,DUM,IXS,IXS,NEQ,MBAND,16,2,NCC,NCC)
C
      CALL ESMBL(GKO,DUM,E,DUM,IXF,IXF,NEQ,MBAND,16,2,NCC,NCC)
```

```
C
      CALL ESMBL(GKO,DUM,C,DUM,IXS,IXF,NEQ,MBAND,16,2,NCC,NCC)
C
      CALL ESMBL(GKO,DUM,C,DUM,IXF,IXS,NEQ,MBAND,16,2,NCC,NCC)
C
C
  500 CONTINUE
C
      IF(NTCSF.EQ.2)  GO TO 630
C
      DO 600 I=1,NEQ
      DO 550 J=1,MBAND
      GHIJ = GH(I,J)
      IF(GHIJ.EQ.0.0) GO TO 550
      GH(I,J) = C3*GHIJ
      GKO(I,J) = GKO(I,J)+GH(I,J)
  550 CONTINUE
  600 CONTINUE
C
C
C      COMPUTE BOUND OF GH(I,J)
C
      CALL PRFIL(GH,MBD,NEQ,MBAND)
C
  630 CONTINUE
      IF(NTCSF.EQ.1) GO TO 900
C
      IF(ICONST.EQ.0) GO TO 750
      DO 700 I=1,NEQ
      DO 650 J=1,MBAND
      GMIJ = GM(I,J)
      IF(GMIJ.EQ.0.0) GO TO 650
      GM(I,J) = C2*GMIJ
      GKO(I,J) = GKO(I,J)+GM(I,J)
  650 CONTINUE
  700 CONTINUE
C
C      COMPUTE BOUND OF GM(I,J)
C
      CALL PRFIL(GM,MBM,NEQ,MBAND)
C
      GO TO 900
C
  750 CONTINUE
C
      DO 800 I=1,NEQ
      GMI = GM(I,1)
      IF(GMI.EQ.0.0) GO TO 800
      GM(I,1) = C2*GMI
      GKO(I,1) = GKO(I,1)+GM(I,1)
  800 CONTINUE
```

```
C
  900 CONTINUE
C
      IF(NVIS.EQ.0) GO TO 950
      READ(5,1001) (HEAD(I),I=1,20)
      WRITE(6,1001) (HEAD(I),I=1,20)
      DO 920 I=1,NVIS
      READ(5,1002) INOD,IDIR,VISC(I)
      IDVIS = ID(INOD,IDIR)
      IVIS(I) = IDVIS
      WRITE(6,1002) INOD,IDIR,VISC(I)
  920 CONTINUE
 1001 FORMAT(20A4)
 1002 FORMAT(2I5,E10.3)
  950 CONTINUE
C
C     READ INITIAL EFFECTIVE STRESS AND PRESSURE
C
C     WRITE(1,*) *BEFORE READING RINTF*
      IF(ISFG.EQ.1.OR.ISFG.EQ.2) CALL RINTS(SRT,NUMEL,MXIP)
      IF(ISFG.EQ.1.OR.ISFG.GE.3) CALL RINTF(PRF,NUMEL,MXIP)
C     WRITE(1,*) *AFTER READING RINTF*
C
C     READ BASE ACCELERATION OR DYNAMIC LOAD
C
      IF(NFG.NE.3)
     * CALL RLOAD(TS,GACL,ID,DYL,IEQH,CINT,TD,UNV,
     * NUMNP,NEQ,NUMLH)
C
C     READ SPECIFIED VELOCITY INPUT
C
      IF(IVEL.NE.O) CALL RVELB(IVEL,NUMVH,NUMVTP,
     * ID,KVEL,KEQH,VINT,TV,SVEL,NUMNP,
     * NVTYPE,DTXV)
C
C
      REWIND 3
C
      RETURN
      END
      SUBROUTINE MASSM(EMS,EMC,EMF,EN,WIWJ,ICONST,RO,ROF,POR,NC,NCC,
     *   NINT,KS,KF,NTCSF)
      DIMENSION EMS(16,1),EMC(16,1),EMF(16,1),EN(8,1),WIWJ(1)
C     EMA EN,WIWJ,NINT
      ROFDP = ROF/POR
      NN = NCC
      IF(ICONST.EQ.0) NN = 1
C
      CALL SZERO(EMS,16*NN)
      CALL SZERO(EMF,16*NN)
      IF(ICONST.EQ.0) GO TO 50
      CALL SZERO(EMC,16*NCC)
C
   50 CONTINUE
C
      DO 200 L=1,NINT
C
      DO 150 I=1,NC
      B = EN(I,L)
      DO 100 J=1,NC
      IF(ICONST.NE.0) NN = J
      EMS(I,NN) = EMS(I,NN)+EN(J,L)*B*WIWJ(L)
  100 CONTINUE
  150 CONTINUE
  200 CONTINUE
C
      DO 400 I=1,NC
      NN = NC
      MM = NC
      IF(ICONST.EQ.0) NN = 1
      IF(ICONST.EQ.0) MM = 0
      DO 300 J=1,NN
      EIJ = EMS(I,J)
      EMS(I,J) = RO*EMS(I,J)
      EMS(I+NC,J+MM) = EMS(I,J)
      IF(KF.EQ.1.OR.NTCSF.EQ.2) GO TO 300
C
      EMF(I,J) = ROFDP*EIJ
      EMF(I+NC,J+MM) = EMF(I,J)
      IF(ICONST.EQ.0) GO TO 300
C
      EMC(I,J) = ROF*EIJ
      EMC(I+NC,J+NC) = EMC(I,J)
  300 CONTINUE
  400 CONTINUE
C
      RETURN
      END
      SUBROUTINE ELH(H,EN,WIWJ,NINT,RXX,RYY,RXY,NCC,NC)
      DIMENSION H(16,NCC),EN(8,1),WIWJ(1)
C     EMA EN,WIWJ,NINT
      CALL SZERO(H,16*NCC)
      DO 50 II=1,NINT
      DO 50 I=1,NC
      EII = EN(I,II)
      DO 50 J=1,NC
      H(I,J) = H(I,J)+EII*EN(J,II)*WIWJ(II)
   50 CONTINUE
C
      DO 80 I=1,NC
      DO 80 J=1,NC
      HIJ = H(I,J)
      H(I,J) = RXX*HIJ
      H(I+NC,J+NC) = RYY*HIJ
C
      IF(RXY.EQ.0.0) GO TO 80
      H(I,J+NC) = RXY+HIJ
      H(J+NC,I) = H(I,J+NC)
   80 CONTINUE
      RETURN
      END
      SUBROUTINE ECTTX(B,EN,XBAR,IP,NC,NCC,TT,EE,E,C,NINT,WIWJ,CM)
      DIMENSION B(2,8,1),EN(8,1),WIWJ(1),XBAR(1),TT(16,1),
     *   EE(16,NCC),E(16,NCC),C(16,NCC)
C     EMA B,EN,XBAR,TT,NINT,WIWJ
C
      CALL SZERO(E,16*NCC)
C
      DO 500 II=1,NINT
C
      DO 300 I=1,NC
C
      IF(IP.LT.3) GO TO 200
      IF(IP.EQ.3) TT(I,II) = B(1,I,II)+EN(I,II)/XBAR(II)
      IF(IP.EQ.4) TT(I,II) = B(1,I,II)+2.*EN(I,II)/XBAR(II)
      GO TO 250
C
  200 TT(I,II) = B(1,I,II)
  250 TT(I+NC,II) = B(2,I,II)
C
  300 CONTINUE
C
      DO 400 I=1,NCC
      DO 400 J=1,NCC
  400 E(I,J) = E(I,J)+TT(I,II)*TT(J,II)*WIWJ(II)*CM
C
  500 CONTINUE
C
      DO 600 I=1,NCC
      DO 600 J=1,NCC
      E(J,I) = E(I,J)
      EE(I,J) = E(I,J)
      C(I,J) = E(I,J)
      EE(J,I) = EE(I,J)
      C(J,I) = C(I,J)
  600 CONTINUE
C
      RETURN
      END
      SUBROUTINE TWOD(XX,YY,MIDSID,INTR,INTS,NEL,B,EN,EJ,XBAR,YBAR,
     *  NC,WIWJ,IP,ISRT)
      COMMON /GAUSQ/ ZETA(4,4),W(4,4)
      DIMENSION XX(1),YY(1),MIDSID(1),B(2,8,1),EN(8,1),EJ(1),
     *   XBAR(1),YBAR(1),NOD5(4),P(2,8),XJ(2,2),XJI(2,2),WIWJ(1)
C     EMA B,EN,XBAR,YBAR,WIWJ
C
C     INPUT :
C     XX(8) = X COORDINATES OF NODES
C     YY(8) = Y COORDINATES OF NODES
C     MIDSID(4) = 0 NO MIDSIDE NODE
C                 1 SPECIFIED MIDSIDE NODE
C     INTR = NUMBER OF INTEGRATION NODES ALONG LOCAL R-AXIS
C     INTS = NUMBER OF INTEGRATION NODES ALONG LOCAL S-AXIS
C     NEL = ELEMENT NUMBER
C     NC = NUMBER OF ELEMENT NODES
C
C     OUTPUT :
C     B(2,8,INTR*INTS) = DERIVATIVES OF SHAPE FUNCTION WITH RESPECT
C                        TO X AND Y AT INTEGRATION NODES
C     EN(8,INTR*INTS) = SHAPE FUNCTION AT INTEGRATION NODES
C     EJ(INTR*INTS) = DETERMINANT OF JACOBIAN AT INTEGRATION NODES
C     XBAR(INTR*INTS) = X COORDINATES AT INTEGRATION NODES
C     YBAR(INTR*INTS) = Y COORDINATES AT INTEGRATION NODES
C
      II = ISRT
      DO 100 IR=1,INTR
      DO 100 IS=1,INTS
      II = II+1
      R = ZETA(IR,INTR)
      S = ZETA(IS,INTS)
      WIWJ(II) = W(IR,INTR)*W(IS,INTS)
      NC = MIDSID(1)+MIDSID(2)+MIDSID(3)+MIDSID(4)+4
C
C
      CALL SHAPE(R,S,EN(1,II),D1,MIDSID,P,XX,YY,NC,IP,D2,D3,0)
C
C     EVALUATE THE JACOBIAN MATRIX AT POINT(R,S)
C
      DO 40 I=1,2
      DUMX = 0.0
      DUMY = 0.0
      DO 30 K=1,NC
      DUMX = DUMX+P(I,K)*XX(K)
      DUMY = DUMY+P(I,K)*YY(K)
   30 CONTINUE
      XJ(I,1) = DUMX
      XJ(I,2) = DUMY
   40 CONTINUE
C
C     COMPUTE THE DETERMINANT OF JACOBIAN MATRIX AT POINT (R,S)
C
      EJ(II) = XJ(1,1)*XJ(2,2)-XJ(2,1)*XJ(1,2)
      DUM = ABS(EJ(II))
      IF(DUM.GT.1.0E-8) GO TO 50
      WRITE(6,55) NEL
   55 FORMAT(1H0,*---- ERROR ZERO JACOBIAN FOR ELEMENT = *,I5)
      STOP
   50 CONTINUE
C
C     INVERSE OF JACOBIAN MATRIX
C
      DUM = 1./EJ(II)
      XJI(1,1) = XJ(2,2)*DUM
      XJI(1,2) =-XJ(1,2)*DUM
      XJI(2,1) =-XJ(2,1)*DUM
      XJI(2,2) = XJ(1,1)*DUM
C
C     CALCULATE B-MATRIX,XBAR, AND YBAR AT POINT (R,S)
C
      XBAR(II) = 0.0
      YBAR(II) = 0.0
      DO 60 I=1,NC
      XBAR(II) = XBAR(II)+EN(I,II)*XX(I)
   60 YBAR(II) = YBAR(II)+EN(I,II)*YY(I)
      DO 65 I=1,NC
      B(1,I,II) = 0.0
   65 B(2,I,II) = 0.0
      DO 80 K=1,NC
      DO 70 I=1,2
      B(1,K,II) = B(1,K,II)+XJI(1,I)*P(I,K)
   70 B(2,K,II) = B(2,K,II)+XJI(2,I)*P(I,K)
   80 CONTINUE
      IF(IP.LT.3) GO TO 90
      IF(IP.EQ.3) WIWJ(II) = XBAR(II)*WIWJ(II)*EJ(II)
      IF(IP.EQ.4) WIWJ(II) = 0.5*XBAR(II)*XBAR(II)*WIWJ(II)*EJ(II)
      GO TO 100
   90 WIWJ(II) = WIWJ(II)*EJ(II)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE SHAPE(R,S,EN,WT,MIDSID,P,XX,YY,NC,IP,RD,FC,NS)
      DIMENSION EN(1),P(2,1),MIDSID(4),NOD5(4),XX(1),YY(1),RD(1),
     *   XJ(1,2),IPERM(4)
C     EMA EN
      DATA IPERM/2,3,4,1/
C
C
      RP = 1.0+R
      SP = 1.0+S
      RM = 1.0-R
      SM = 1.0-S
      R2 = 1.0-R*R
      S2 = 1.0-S*S
C
C     INTERPOLATION FUNCTION AND THEIR DERIVATIVES
```

```
C     4-NODE ELEMENT
C
      EN(1) = 0.25*RP*SP
      EN(2) = 0.25*RM*SP
      EN(3) = 0.25*RM*SM
      EN(4) = 0.25*RP*SM
      P(1,1) = 0.25*SP
      P(1,2) = -P(1,1)
      P(1,3) = -0.25*SM
      P(1,4) = -P(1,3)
      P(2,1) = 0.25*RP
      P(2,2) = 0.25*RM
      P(2,3) = -P(2,2)
      P(2,4) = -P(2,1)
      IF(NC.EQ.4) GO TO 20
C
C     ADD DEGREES OF FREEDOM IN EXCESS OF FOUR
C
      JJ = 0
      DO 1 I=1,4
      IF(MIDSID(I).EQ.0) GO TO 1
      JJ = JJ+1
      NOD5(JJ) = I+4
    1 CONTINUE
      I = 0
      NND5 = NC-4
    2 I = I+1
      IF(I.GT.NND5) GO TO 10
      NN = NOD5(I)-4
      GO TO (5,6,7,8) NN
    5 EN(5) = 0.5*R2*SP
      P(1,5) = -R*SP
      P(2,5) = 0.5*R2
      GO TO 2
    6 EN(6) = 0.5*RM*S2
      P(1,6) = -0.5*S2
      P(2,6) = -RM*S
      GO TO 2
    7 EN(7) = 0.5*R2*SM
      P(1,7) = -R*SM
      P(2,7) = -0.5*R2
      GO TO 2
    8 EN(8) = 0.5*RP*S2
      P(1,8) = 0.5*S2
      P(2,8) = -RP*S
      GO TO 2
   10 IH = 0
   11 IH = IH+1
      IF(IH.GT.NND5) GO TO 20
      IN = NOD5(IH)
      I1 = IN-4
      I2 = IPERM(I1)
      EN(I1) = EN(I1)-0.5*EN(IN)
      EN(I2) = EN(I2)-0.5*EN(IN)
      EN(IH+4) = EN(IN)
      DO 15 J=1,2
      P(J,I1) = P(J,I1)-0.5*P(J,IN)
      P(J,I2) = P(J,I2)-0.5*P(J,IN)
   15 P(J,IH+4) = P(J,IN)
      GO TO 11
   20 CONTINUE
C
      IF(NS.EQ.0) RETURN
      I = NS
      IF(I.EQ.1) LL = 2
      IF(I.EQ.2) LL = 1
C
      DUM1 = 0.0
      DUM2 = 0.0
      DO 23 K=1,NC
      DUM1 = DUM1+P(LL,K)*XX(K)
      DUM2 = DUM2+P(LL,K)*YY(K)
   23 CONTINUE
C
      XJ(1,1) = DUM1
      XJ(1,2) = DUM2
C
      IF(IP.LT.3) GO TO 26
      XIBAR = 0.0
C
      DO 25 K1=1,NC
   25 XIBAR = XIBAR+EN(K1)*XX(K1)
C
   26 IF(IP.LT.3) XIBAR = 1.0
      IF(IP.EQ.3) XIBAR = XIBAR
      IF(IP.EQ.4) XIBAR = XIBAR*XIBAR
      FAC = FC*WT*XIBAR
      IF(I.EQ.2) FAC = -FAC
      JJ = 0
C
      DO 27 J=1,NC
      JJ = JJ+1
      RD(J) = RD(J)+EN(JJ)*XJ(1,2)*FAC
      RD(J+NC) = RD(J+NC)-EN(JJ)*XJ(1,1)*FAC
   27 CONTINUE
C
      RETURN
      END
      SUBROUTINE PRFIL(A,MB,NEQ,MBAND)
      DIMENSION A(NEQ,1),MB(1)
C     EMA A,MB
      DO 300 N=1,NEQ
      NI = 1
      NJ = 1
      DO 100 M=2,MBAND
      IF(A(N,M).NE.0.0) NI = M
      IF(N-M) 100,80,80
   80 IF(A(N-M+1,M).NE.0.0) NJ = M
  100 CONTINUE
      MB(N) = NI
      MB(N+NEQ) = NJ
  300 CONTINUE
      RETURN
      END
      SUBROUTINE RINTF(PRF,NUMEL,MXIP)
      DIMENSION PRF(MXIP,1)
C     EMA PRF
      WRITE(6,2010)
 2010 FORMAT(
     .*INITIAL PORE FLUID PRESSURES *//
     .* ELEMENT    PORE   *//
     .*  NUMBER  PRESSURE *//)
      DO 100 J=1,NUMEL
      READ (5,1000) (PRF(I,J),I=1,MXIP)
      WRITE(6,2000) J,(PRF(I,J),I=1,MXIP)
  100 CONTINUE
 1000 FORMAT(8F10.0)
 2000 FORMAT(I6,2X,8E10.3)
      RETURN
      END
      SUBROUTINE RINTS(SRT,NUMEL,MXIP)
      DIMENSION SRT(4,MXIP,1)
C     EMA SRT
C
C     READ EFFECTIVE INITIAL STRESS
C
      WRITE(6,2000)
      DO 100 I=1,NUMEL
      DO 50 K=1,MXIP
      READ(5,1000) (SRT(J,K,I),J=1,4)
      WRITE(6,2010) I,(SRT(J,K,I),J=1,4)
   50 CONTINUE
  100 CONTINUE
C
 1000 FORMAT(4F10.0)
 2000 FORMAT(/*  INITIAL STRESS DATA   */
     .*      ELNO       SIGR       SIGZ       SIGT       SIGRZ */)
 2010 FORMAT(I10,4E10.3)
      RETURN
      END
      SUBROUTINE RLOAD(TS,GACL,ID,DYL,IEQH,CINT,TD,
     *   UNV,NUMNP,NEQ,NUMLH)
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      COMMON /ACCLF/ MTYPE,NUMAP,DTXA
      COMMON /PRELF/ NUMLP,NUMLX,NUMTP,NTYPE,DTXX
      DIMENSION TS(1),GACL(1),ID(NUMNP,1),UNV(1),
     *   DYL(NUMLH,1),IEQH(3,1),CINT(1),TD(1),IHED(40)
C     EMA TS,GACL,ID,DYL,IEQH,CINT,TD,UNV
      IF(NFG.NE.1) GO TO 300
C
      IF(ICONST.EQ.0) GO TO 35
C
C     BASE ACCELERATION ( NFG = 1 )
C       INITIALIZE UNIT VECTOR CORRESPONDING TO HORIZONTAL DOFS
C
      CALL EZERO(UNV,NEQ)
C
      DO 30 I=1,NUMNP
      DO 20 J=1,2
      II = J+J-1
      JJ = ID(I,II)
      IF(JJ) 20,20,16
   16 UNV(JJ) = 1.0
   20 CONTINUE
   30 CONTINUE
C
C
C      READ BASE ACCELERATION
C
   35 CONTINUE
      READ (5,1004) IHED
      READ (5,1012) TSCAL,ASCAL
      WRITE(6,2008) IHED,TSCAL,ASCAL
      IF(MTYPE.EQ.0) GO TO 41
C
      READ(5,1008) (TS(I),GACL(I),I=1,NUMAP)
      DO 40 I=1,NUMAP
      TS(I) = TS(I)*TSCAL
   40 GACL(I) = GACL(I)*ASCAL
      WRITE(6,2009) (N,TS(N),GACL(N),N=1,NUMAP)
      GO TO 44
C
   41 CONTINUE
      READ(5,1013) (GACL(I),I=1,NUMAP)
      IF(ASCAL.EQ.1.0) GO TO 43
      DO 42 I=1,NUMAP
   42 GACL(I) = ASCAL*GACL(I)
   43 CONTINUE
      WRITE(6,2022) (GACL(I),I=1,NUMAP)
   44 CONTINUE
C
      RETURN
C
  300 CONTINUE
```

```
C
C       READ DYNAMIC LOADS
C       PRESSURE LOAD ( NFG = 2 OR 4 )
C
      CALL EZERO(DYL,NUMLH*NUMTP)
      CALL EZERO(IEQH,3*NUMLP)
      CALL EZERO(CINT,NUMLP)
C
      READ(5,1004) IHED
      WRITE(6,2001) IHED
C
      DO 350 N=1,NUMLP
      READ(5,1000) JNOD,JDOF,IEQH(2,N),CINT(N)
      WRITE(6,2000) JNOD,JDOF,IEQH(2,N),CINT(N)
      IEQH(1,N) = ID(JNOD,JDOF)
      IF(IP.NE.3) GO TO 350
      IEQH(3,N) = JNOD
  350 CONTINUE
C
C
      IF(NTYPE.EQ.0) GO TO 380
      READ(5,1012) (TD(I),I=1,NUMTP)
      GO TO 384
  380 CONTINUE
C     WRITE(1,*) *BEFORE DO 382*
      DO 382 I=1,NUMTP
  382 TD(I) = DTXX*(I-1)
  384 DO 400 I=1,NUMLH
      READ(5,1012) (DYL(I,J),J=1,NUMTP)
C     WRITE(1,*) *AFTER READING LOAD*
      READ(5,1001)
  400 CONTINUE
C
      WRITE(6,2202)
 2202 FORMAT(
     .*   TIME         PRESSURE *//)
      DO 420 I=1,NUMTP
      WRITE(6,2002) TD(I),(DYL(J,I),J=1,NUMLH)
  420 CONTINUE
      RETURN
 1000 FORMAT(3I5,E10.3)
 1001 FORMAT( A1 )
 1004 FORMAT(40A2)
 1008 FORMAT(12F6.0)
 1012 FORMAT(7E10.0)
 1013 FORMAT(8F9.0)
C
 2000 FORMAT(3I5,2X,E10.3)
 2001 FORMAT( 40A2//
     .*  NODE IDOF LHNO    CINT */)
 2002 FORMAT(E10.3,5X,8(E10.3))
 2008 FORMAT(1H1/40A2/
     . * SCALE FACTOR FOR TIME.................. *,F10.3/
     . * SCALE FACTOR FOR ACCELERATION.......... *,F10.3//)
 2009 FORMAT(/5X,*POINT*,11X,*TIME*,11X,*ACCEL*/(5X,I5,2F15.6))
 2022 FORMAT(1X,/,* ACCELERATION -----*/(1X,8(F10.5,1X)))
C
      END
      SUBROUTINE RVELB(IVEL,NUMVH,NUMVTP,ID,
     * KVEL,KEQH,VINT,TV,SVEL,NUMNP,NVTYPE,DTXV)
      DIMENSION ID(NUMNP,1),
     *  KVEL(1),KEQH(1),VINT(1),TV(1),SVEL(NUMVTP,1),IHED(40)
C     EMA ID,KVEL,KEQH,VINT,TV,SVEL
C
C     READ SPECIFIED VELOCITY TIME HISTORY
C
      READ(5,1004) IHED
      WRITE(6,1004) IHED
 1004 FORMAT(40A2)
C
      DO 100 N=1,IVEL
      READ(5,1006) JNOD,JDOF,KEQH(N),VINT(N)
      WRITE(6,1006) JNOD,JDOF,KEQH(N),VINT(N)
      KVEL(N) = ID(JNOD,JDOF)
  100 CONTINUE
 1006 FORMAT(3I5,F10.0)
C
      IF(NVTYPE.EQ.1) READ(5,1008) (TV(I),I=1,NUMVTP)
      IF(NVTYPE.EQ.1) GO TO 200
      TV(1) = 0.0
      DO 150 I=2,NUMVTP
      TV(I) = TV(I-1)+DTXV
  150 CONTINUE
  200 WRITE(6,1008) (TV(I),I=1,NUMVTP)
 1008 FORMAT(7E10.3)
C
      DO 300 I=1,NUMVH
      READ(5,1008) (SVEL(J,I),J=1,NUMVTP)
      WRITE(6,1008) (SVEL(J,I),J=1,NUMVTP)
  300 CONTINUE
  400 CONTINUE
C
      RETURN
      END
/
```

```
C FTN7X,L
C $EMA /WAREA/,/ZAREA/
C     PROGRAM DPSM3(5)
      SUBROUTINE DPSM3
C     INTEGER*4 N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,
C    * N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28
C     INTEGER*4 N29,N30,N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B
      COMMON /WAREA/ A(60000)
      COMMON /NPARM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,
     *   N14,N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,
     *   N28,N29,N30,N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B
      COMMON /ZAREA/ Z(1500)
      COMMON /LPARM/ L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,MAXIP,MXIP,MXSH,
     *   LBBB,LDDD,NELS
      COMMON /MMBDP/ NUMNP,NUMEL,NUMMAT,NVIS,NSKEW,NEQ,MBAND
      COMMON /PRELF/ NUMLP,NUMLH,NUMTP,NTYPE,DTXX
      COMMON /VELLF/ NUMVEL,NUMVH,NUMVTP,NVTYPE,DTXV
C
C     WRITE(6,*) N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
C     WRITE(6,*) N11,N12,N13,N14,N15,N16,N17,N18,N19,N20
C     WRITE(6,*) N21,N22,N23,N24,N25,N26,N27,N28,N29,N30
C     WRITE(6,*) N31,N32,N33,N34,N35,N36,N37,N38,N5A,N5B
C     WRITE(6,*) * LPARAMETERS *
C     WRITE(6,*) L1,L2,L3,L4,L5,L6,L7,L8,L9,L10
C     WRITE(6,*) MAXIP,MXIP,MXSH,LBBB,LDDD,NELS
      CALL SOLVE(NEQ,NUMLH,NUMNP,A(N2),A(N3),A(N4),A(N5B),A(N6),A(N7),
     * A(N8),A(N9),A(N10),A(N11),A(N12),A(N13),A(N14),A(N15),A(N16),
     * A(N17),A(N18),A(N19),A(N20),A(N21),A(N1),A(N22),A(N23),A(N24),
     * A(N25),A(N26),A(N27),A(N28),NUMEL,A(N29),A(N30),A(N31),A(N32),
     * A(N33),A(N34),A(N35),A(N36),A(N37),A(N38),NUMVTP,A(N5),
     * A(N5A),MBAND,Z(L1),Z(L2),Z(L3),Z(L4),Z(L5),Z(L6),Z(L1),
     * Z(L7),Z(L8),Z(L9),Z(L8),MAXIP,MXIP,MXSH,LBBB,LDDD,NELS)
C
C     CALL SEGRT
      RETURN
C
      END
      SUBROUTINE POPS(PF,TT,DU,DW,CM,NCC)
      DIMENSION TT(1),DU(1),DW(1)
C     EMA PF,TT
C
C     COMPUTE PORE PRESSURE INCREMENT
C
      DPF = 0.0
      DO 100 I=1,NCC
      DPF = DPF+CM*(TT(I)*DU(I)+TT(I)*DW(I))
  100 CONTINUE
C
C     UPDATE PORE PRESSURE
C
```

```
      PF = PF+DPF
      RETURN
      END
      SUBROUTINE RESLF(TT,PF,WIWJ,NCC,RS,RF)
      DIMENSION TT(1),RS(1),RF(1)
C     EMA TT,WIWJ
C
C     SOLID AND FLUID PHASE ELEMENT LOAD VECTOR DUE TO PORE PRESSURE
C
      WPF = -WIWJ*PF
      DO 100 I=1,NCC
      RS(I) = RS(I)+WPF*TT(I)
      RF(I) = RF(I)+WPF*TT(I)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE CALTT(TT,B,EN,XBAR,NC,IP,MAXIP)
      DIMENSION B(2,8,1),EN(8,1),TT(16,1)
C     EMA TT,B,EN,XBAR
      DO 500 II=1,MAXIP
      DO 300 I=1,NC
      IF(IP.LT.3) GO TO 200
      IF(IP.EQ.3) TT(I,II) = B(1,I,II)+EN(I,II)/XBAR(II)
      IF(IP.EQ.4) TT(I,II) = B(1,I,II)+2.*EN(I,II)/XBAR(II)
      GO TO 250
  200 TT(I,II) = B(1,I,II)
  250 TT(I+NC,II) = B(2,I,II)
  300 CONTINUE
  500 CONTINUE
      RETURN
      END
C
C $EMA /OUTPS/,/OUTHS/
      SUBROUTINE SOLVE(NEQ,NUMLH,NUMNP,XA,YA,PROP,GKO,GK,GH,GM,MB,
     *   MBM,MBD,AO,OAO,RLV,UI,VA,DISP,VEL,ACC,TS,GACL,ID,DYL,IEQH,CINT,
     *   TD,UNV,SRT,PRF,NUMEL,DAAA,DBBB,DDDD,IVIS,VISC,KVEL,KEQH,
     *   VINT,TV,SVEL,NUMVTP,ISKEW,ACS,MBAND,NINT,B,EN,XBAR,YBAR,
     *   WIWJ,BBB,TT,PSA,PFA,DDD,MAXIP,MXIP,MXSH,LBBB,LDDD,NELS)
      COMMON /BATCH/ IBATCH
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      COMMON /GCALP/ NCYCL,DT,NUPDAT,ITER
      COMMON /MMBDP/ NUMNX,NUMEX,NUMMAT,NVIS,NSKEW,NEQX,MBANDX
      COMMON /ACCLF/ MTYPE,NUMAP,DTXA
      COMMON /PRELF/ NUMLP,NUMLX,NUMTP,NTYPE,DTXX
      COMMON /VELLF/ IVEL,NUMVH,NUMVTX,NVTYPE,DTXV
      COMMON /OUTPS/ NPFL,NDC,NSG,NPRINT,NPEL,NPMT,NPRT(1500),NPM(1500)
      COMMON /OUTHS/ NTHS,NHPEL,NHPMT,NHPRT(100),NHPM(100)
      COMMON /INTCN/ AA,AA1,AA2,BB,BB1,BB2,A1,A2,A3,B1,B2,B3,
     *   C1,C2,C3,TETA,BETA,GAMA,ALPA
      COMMON /EDT/ NEL,IXS(16),IXF(16),NP(8),MIDSID(4),INN(3),
     *   ISF(8),NC,NCC
```

```
C
      DIMENSION XA(1),YA(1),
     *   GKO(NEQ,MBAND),GH(NEQ,1),GM(NEQ,1),MB(1),MBM(1),MBD(1),
     *   AO(1),OAO(1),RLV(1),DU(16),DW(16),UI(1),RS(16),RF(16),STRN(4),
     *   STRS(4),DP(4,4),EST(16,16),VA(1),DISP(1),VEL(1),ACC(1),TS(1),
     *   GACL(1),ID(NUMNP,1),DYL(NUMLH,1),IEQH(3,1),CINT(1),TD(1),
     *   PROP(30,1),UNV(1),SS(4,16),SRT(4,MXIP,1),PRF(MXIP,1),SWI(4),
     *   IVIS(1),VISC(1),KVEL(1),KEQH(1),VINT(1),TV(1),SVEL(NUMVTP,1),
     *   ISKEW(5,1),ACS(2,1),LTIME(15),GK(NEQ,MBAND),
     *   STRSO(4),B(2,8,MAXIP),EN(8,MAXIP),XBAR(MAXIP),YBAR(MAXIP),
     *   WIWJ(MAXIP),BBB(LBBB),TT(16,MAXIP),
     *   PSA(MXSH,MXIP),PFA(MXIP),DDD(LDDD),AAA(58),DAAA(58,NELS),
     *   DBBB(LBBB,NELS),DDDD(LDDD,NELS,1),EDISP(16),ESTRN(4,16)
C     EMA XA,YA,PROP,GKO,GK,GH,GM,MB,MBM,MBD,AO,OAO,RLV,UI,VA,DISP,
C    *   VEL,ACC,TS,GACL,ID,DYL,IEQH,CINT,TD,UNV,SRT,PRF,DAAA,DBBB,
C    *   DDDD,IVIS,VISC,KVEL,KEQH,VINT,TV,SVEL,ISKEW,ACS,NINT,B,EN,
C    *   XBAR,YBAR,WIWJ,BBB,TT,PSA,PFA,DDD
C
      EQUIVALENCE(NEL,AAA(1))
C
C
      CALL EZERO(OAO,NEQ)
      NH = MBAND-1
      NCL = 0
      LSTRT = 0
      TM = 0.0
      NFIRST = 0
      NDSK1 = 10
      NDSK2 = 11
      KUPDAT = 0
      NFSTF = 1
      NTER = 0
      IEND = ITER
      IF(ITER.NE.0) ITER = 1
      ISAS = IEND-1
      MMM = 1
      NNN = 0
      MCO = 0
      ITENTH = 10
      IPER = 0
      CALL EZERO(UI,NEQ)
      CALL EZERO(DISP,NEQ)
      CALL EZERO(VEL,NEQ)
      CALL EZERO(ACC,NEQ)
C
      IF(NELS.EQ.NUMEL) CALL TRANS(3,DAAA,NUMEL,58,AAA)
C
  100 CONTINUE
      IF(NTER.EQ.1) GO TO 120
      NCL = NCL+1
C     CALL FTIME(LTIME)
```

```
C     WRITE(1,1) NCL,(LTIME(I),I=1,15)
C    1 FORMAT(*STARTING CYCLE = *,I5/15A2)
      IF(IBATCH.EQ.1) GO TO 3220
      RPER = NCL*100./NCYCL
      IPER = IFIX(RPER)
      IF(IPER.NE.ITENTH) GO TO 3220
      PRINT 3210,ITENTH,NCL
 3210 FORMAT(1X,"FINISHED WITH ",I3,"% OF THE STEPS AT STEP # ",
     .I5)
      ITENTH = ITENTH + 10
 3220 CONTINUE
      TM = TM+DT
  120 CONTINUE
      IF(NFIRST.EQ.0) GO TO 162
      KUPDAT = KUPDAT+1
      IF(KUPDAT.LT.NUPDAT) GO TO 164
  162 NFSTF = 1
      KUPDAT = 0
      GO TO 165
  164 NFSTF = 0
  165 CONTINUE
      NSK = 1
      NSU = 1
      NSR = 1
      IF(NELS.EQ.NUMEL) GO TO 182
      REWIND 3
      REWIND 4
      REWIND NDSK1
      REWIND NDSK2
 182  CONTINUE
      IF(NLNR.EQ.0.AND.NFIRST.NE.0) GO TO 190

      IF(NFSTF.NE.1) GO TO 190
C     WRITE(6,*) "BEFORE EDPLV"
      CALL TDPLV(GKO,GK,NEQ,MBAND)
C     WRITE(1,*) *AFTER EDPLV*
C
  190 CONTINUE
      CALL EZERO(RLV,NEQ)
      ICNT = 1
      IHCNT = 1
      IF(NTER.EQ.1) GO TO 210
      IF(MMM.NE.0) GO TO 210
      WRITE(MSTR,2010) TM-DT
      WRITE(MSTR,2020)
  210 CONTINUE

      DO 330 NUM = 1,NUMEL
      RELP = FLOAT(NUM)
      IF(NELS.EQ.0) READ (3) AAA
      IF(NELS.EQ.NUMEL) CALL WRID3(DAAA(1,NUM),58,AAA,0)
      MAT = ISF(1)
      KS = ISF(3)
      KF = ISF(4)
C
C     GOOD FOR CYBER
C
      IF(NELS.EQ.0) READ (4) BBB
      IF(NELS.EQ.NUMEL) CALL WREAD(DBBB(1,NUM),LBBB,BBB,0)
      EMNO = PROP(1,MAT)
      MNO = INT(EMNO)
      TENS = PROP(2,MAT)
      ITEN = INT(TENS)
      STIFAC = PROP(3,MAT)
      SHEFAC = PROP(4,MAT)
      PMN = PROP(5,MAT)
      BK = PROP(6,MAT)
      G = PROP(7,MAT)
      CM = PROP(26,MAT)
      NMPK = 0
      IF(EMNO.NE.0.0) NMPK = 1
      IF(NFIRST.EQ.0) GO TO 230
C
C     GOOD FOR CYBER
C
      IF(NELS.EQ.0) READ (NDSK1) DDD
      IF(NELS.EQ.NUMEL) CALL WREAD(DDDD(1,NUM,NDSK1-9),LDDD,DDD,0)
      GO TO 240
  230 CONTINUE
      IF(NLNR.NE.0) GO TO 235
      CALL EZERO(PSA,MXSH*MXIP)
      IF(ISFG.EQ.0.OR.ISFG.GE.3) GO TO 240
      DO 232 I=1,4
      DO 232 J=1,MXIP
  232 PSA(I,J) = SRT(I,J,NUM)
      GO TO 240
C 235 CALL INITZ(PSA,NUM,SRT,PROP,MAT,NINT)
  235 CALL EZERO(PSA,MXSH*MXIP)
      IF(ISFG.EQ.0.OR.ISFG.GE.3) GO TO 240
      DO 236 I=1,4
      DO 236 J=1,MXIP
  236 PSA(I,J) = -SRT(I,J,NUM)
  240 CONTINUE
C
      IF(NFIRST.NE.0) GO TO 242
      CALL SZERO(DU,NCC)
      CALL SZERO(EDISP,NCC)
      IF(NTCSF.EQ.2) GO TO 245
      CALL SZERO(DW,NCC)
      CALL EZERO(PFA,MXIP)
      IF(ISFG.EQ.0.OR.ISFG.EQ.2) GO TO 245
      DO 241 I=1,MXIP
      PFA(I) = PRF(I,NUM)
  241 CONTINUE
C
      GO TO 245
  242 CONTINUE
      CALL EXTRT(DU,IXS,UI,NCC)
      CALL EXTRT(EDISP,IXS,DISP,NCC)
      IF(NTCSF.NE.2) CALL EXTRT(DW,IXF,UI,NCC)
C
      IF(NSKEW.EQ.0) GO TO 243
      IF(NSU.GT.NSKEW) GO TO 243
      IF(NUM.NE.ISKEW(1,NSU)) GO TO 243
      INX = ISKEW(2,NSU)
      JNX = ISKEW(3,NSU)
      KNX = ISKEW(4,NSU)
      MSF = ISKEW(5,NSU)
      CSC = ACS(1,NSU)
      CSS = -ACS(2,NSU)
      IF(MSF.EQ.1) GO TO 2425
      CALL MSKEW(DU,16,NC,1,INX,JNX,KNX,2,CSC,CSS)
      CALL MSKEW(EDISP,16,NC,1,INX,JNX,KNX,2,CSC,CSS)
 2425 IF(MSF.EQ.2) GO TO 2426
      CALL MSKEW(DW,16,NC,1,INX,JNX,KNX,2,CSC,CSS)
 2426 CONTINUE
      NSU = NSU+1
  243 CONTINUE
  245 CONTINUE
C
      CALL SZERO(RS,NCC)
      IF(NTCSF.NE.2) CALL SZERO(RF,NCC)
      IF(NLNR.NE.O.AND.NFSTF.EQ.1) CALL SZERO(EST,16*16)
C
      DO 290 II=1,NINT
      N = II
      M = II
      IF(MXIP.NE.1) GO TO 250
      N = 1
      M = NINT+1
  250 CONTINUE
      IF(MXIP.EQ.1.AND.II.GT.1) GO TO 9901
      IF(NTER.EQ.1) CALL DUPVX(STRSO,PSA(1,N))
      CALL ELSTN(B(1,1,M),EN(1,M),XBAR(M),DU,STRN,NC,IP)
      CALL ELSTN(B(1,1,M),EN(1,M),XBAR(M),EDISP,ESTRN(1,N),NC,IP)
      IF(NFIRST.EQ.O.AND.NLNR.NE.O) PSA(6,N) = -1000.
      IF(NMPK.EQ.1) CALL SIGN(STRN,4)
C
      IF(NMPK.EQ.O.OR.NLNR.EQ.O) GO TO 9000
      GO TO (9001,9002,9003,9004,9005), NLNR
      WRITE(6,2031)
 2031 FORMAT(32H MATERIAL MODEL IS NOT AVAILABLE  )
      STOP
 9000 CONTINUE
C
C     LINEAR ELASTIC MATERIAL MODEL
C
      CALL MTPK0(PSA(1,N),STRN,BK,G)
      CALL ELAST(BK,G,DP)
      GO TO 9009
 9001 CONTINUE
C
C     DECOUPLED ELASTO-PLASTIC MATERIAL MODEL (DCOUP)
C
C     CALL MTPK1(PSA(1,N),STRN,DP,NFIRST,NFSTF,MNO,BK,G,PROP(1,MAT))
      GO TO 9009
 9002 CONTINUE
C
C     CAP MODEL
C
      WRITE(6,2031)
      STOP
 9003 CONTINUE
C
C     AFWL ENGINEERING MODEL
C
      WRITE(6,2031)
      STOP
 9004 CONTINUE
C
C     UNIAXIAL-STRAIN MATERIAL MODEL (UNIAX)
C
C     CALL MTPK4(PSA(1,N),STRN,DP,NFIRST,NFSTF,MNO,PROP(1,MAT))
      GO TO 9009
 9005 CONTINUE
C
C     ARA2D MATERIAL MODEL
C
      CALL MTPK5(PSA(1,N),STRN,DP,NFIRST,NFSTF,BK,G,PROP(1,MAT),IP)
C
 9009 CONTINUE
C
      IF(NLNR.EQ.0) GO TO 280
      IF(NFSTF.EQ.0) GO TO 280
C     IF(ITEN.NE.0.AND.NLNR.NE.5)
C    *              CALL CUTEN(NMPK,DP,PSA(1,N),IP,STIFAC,SHEFAC,PMN,
C    *   PSA(6,N))
 9901 CONTINUE
      IF(NLNR.EQ.0.OR.NFSTF.EQ.0) GO TO 280
      CALL ELSTF(B(1,1,II),DP,EN(1,II),XBAR(II),WIWJ(II),NC,NCC,EST,IP,
     *  0)
  280 CONTINUE
      IF(MXIP.EQ.1.AND.II.GT.1) GO TO 285
C
```

```
      IF(NTER.NE.1) CALL DUPVX(STRS,PSA(1,N))
      IF(NTER.EQ.1) CALL DUPVT(STRSO,STRS)
      IF(NMPK.EQ.1) CALL SIGN(STRS,4)
C
      IF(ISFG.EQ.0.OR.ISFG.GE.3) GO TO 282
      DO 281 I=1,4
  281 SWI(I) = STRS(I)-SRT(I,N,NUM)
      GO TO 285
  282 CONTINUE
      DO 283 I=1,4
  283 SWI(I) = STRS(I)
  285 CALL RESLS(B(1,1,II),SWI,EN(1,II),XBAR(II),WIWJ(II),NC,RS,IP)
C
  290 CONTINUE
C
      IF(NSKEW.EQ.0) GO TO 288
      IF(NSK.GT.NSKEW) GO TO 288
      IF(NUM.NE.ISKEW(1,NSK)) GO TO 288
      MSF = ISKEW(5,NSK)
      IF(MSF.EQ.1) GO TO 286
      INX = ISKEW(2,NSK)
      JNX = ISKEW(3,NSK)
      KNX = ISKEW(4,NSK)
      CSC = ACS(1,NSK)
      CSS = ACS(2,NSK)
      IF(NTCSF.EQ.2) CALL MSKEW(RS,16,NC,1,INX,JNX,KNX,2,CSC,CSS)
      IF(NLNR.NE.0.OR.NFSTF.NE.0)
     * CALL MSKEW(EST,16,NC,NCC,INX,JNX,KNX,3,CSC,CSS)
  286 NSK = NSK+1
  288 CONTINUE
C
C     CALL ALPHA(EST)
      IF(NLNR.NE.0.AND.NFSTF.NE.0)
     * CALL ESMBL(GK,DUM,EST,DUM,IXS,IXS,NEQ,MBAND,16,2,NCC,NCC)
      IF(NTCSF.EQ.2)
     * CALL ESMBL(DUM,RLV,DUM,RS,IXS,DUM,NEQ,DUM,16,3,NCC,DUM)
C
      IF(KF.EQ.1.OR.NTCSF.EQ.2) GO TO 305
C
      CALL CALTT(TT,B,EN,XBAR,NC,IP,MAXIP)
C
      DO 300 II=1,NINT
      N = II
      M = II
      IF(MXIP.NE.1) GO TO 9905
      N = 1
      M = NINT+1
 9905 CONTINUE
      IF(MXIP.EQ.1.AND.II.GT.1) GO TO 295
      IF(NFIRST.NE.0) CALL POPS(PFA(N),TT(1,M),DU,DW,CM,NCC)
      IF(ISFG.EQ.1.OR.ISFG.EQ.3) GO TO 293
      PWI = PFA(N)
      GO TO 295
  293 PWI = PFA(N)-PRF(N,NUM)
  295 CALL RESLF(TT(1,II),PWI,WIWJ(II),NCC,RS,RF)
C
  300 CONTINUE
C
      IF(NSKEW.EQ.0) GO TO 304
      IF(NSR.GT.NSKEW) GO TO 304
      IF(NUM.NE.ISKEW(1,NSR)) GO TO 304
      MSF = ISKEW(5,NSR)
      INX = ISKEW(2,NSR)
      JNX = ISKEW(3,NSR)
      KNX = ISKEW(4,NSR)
      CSC = ACS(1,NSR)
      CSS = ACS(2,NSR)
      IF(MSF.NE.1) CALL MSKEW(RS,16,NC,1,INX,JNX,KNX,2,CSC,CSS)
      IF(MSF.NE.2) CALL MSKEW(RF,16,NC,1,INX,JNX,KNX,2,CSC,CSS)
      NSR = NSR+1
  304 CONTINUE
C
      CALL ESMBL(DUM,RLV,DUM,RS,IXS,DUM,NEQ,DUM,16,3,NCC,DUM)
      CALL ESMBL(DUM,RLV,DUM,RF,IXF,DUM,NEQ,DUM,16,3,NCC,DUM)
C
  305 CONTINUE
      IF(NTER.EQ.1) GO TO 320
C
C
C     GOOD FOR CYBER
C
      IF(NELS.EQ.0) WRITE (NDSK2) DDD
      IF(NELS.EQ.NUMEL) CALL WREAD(DDDD(1,NUM,NDSK2-9),LDDD,DDD,1)
C
      IF(NCL.EQ.1) GO TO 320
      IF(NTHS.EQ.0) GO TO 308
      IF(NTHS.EQ.1) GO TO 308
      IF(IHCNT.GT.NHPEL) GO TO 306
      IF(NHPRT(IHCNT).NE.NUM) GO TO 306
      IHCNT = IHCNT+1
      CALL STRCL(30,NUM,NMPK,ESTRN,MXIP,MXSH,NLNR,NTCSF,NINT,KS,KF,
     *  SS,PSA,PFA)
  306 CONTINUE
  308 IF(MMM.NE.0) GO TO 320
      IF(ICNT.GT.NPEL) GO TO 320
      IF(NPRT(ICNT).NE.NUM) GO TO 320
      ICNT = ICNT+1
      CALL STRCL(MSTR,NUM,NMPK,ESTRN,MXIP,MXSH,NLNR,NTCSF,NINT,KS,KF,
     *  SS,PSA,PFA)
  320 CONTINUE
  330 CONTINUE
C
```

```
C     WRITE(6,*) "AFTER ELEMENT OPERATION"
C
      IF(NCL.GT.NCYCL) GO TO 950
      IF(NTCSF.EQ.1) GO TO 380
C
      DO 350 I=1,NEQ
  350 VA(I) = AA1*VEL(I)+AA2*ACC(I)
      IF(ICONST.EQ.0) GO TO 360
      CALL MLTPL(GM,RLV,VA,MBM,NEQ)
      GO TO 380
  360 CONTINUE
C
      DO 370 I=1,NEQ
  370 RLV(I) = RLV(I)+GM(I,1)*VA(I)
  380 CONTINUE
      IF(NTCSF.EQ.2) GO TO 400
      DO 390 I=1,NEQ
  390 VA(I) = BB1*VEL(I)+BB2*ACC(I)
      CALL MLTPL(GH,RLV,VA,MBD,NEQ)
  400 CONTINUE
C
      ISAS = ISAS+1
      IF(ISAS.GT.IEND) ISAS = 0
      IF(ITER.EQ.1.AND.ISAS.LT.IEND) NTER = 1
      IF(ITER.EQ.1.AND.ISAS.EQ.IEND) NTER = 2
      CALL EZERO(AO,NEQ)
C
C     BASE ACCELERATION OR DYNAMIC LOAD
C
      IF(NFG.NE.3)
     * CALL DLOAD(TS,GACL,ID,GM,AO,TM,DYL,IEQH,CINT,TD,XA,UNV,MBM,
     * LSTRT,NUMNP,NEQ,NUMLH,AA,BB)
C     WRITE(6,*) "AFTER DLOAD"
C
      DO 410 I=1,NEQ
  410 RLV(I) = RLV(I)+OAO(I)+TETA*(AO(I)-OAO(I))
      IF(IVEL.NE.0) CALL VELBC(NLNR,NFIRST,IVEL,NUMVH,NUMVTP,NEQ,
     * MBAND,ID,GK,GKO,RLV,VEL,ACC,KVEL,KEQH,VINT,TV,SVEL,TM,NUMNP,
     * NVTYPE,DTXV)
      IF(NLNR.EQ.0.AND.NFIRST.NE.0) GO TO 415
C
      IF(NFSTF.EQ.0) GO TO 415
      IF(NVIS.EQ.0) GO TO 414
      DO 413 I=1,NVIS
      K = IVIS(I)
      GK(K,1) = GK(K,1)+BB*VISC(I)
  413 CONTINUE
  414 CONTINUE
C
      CALL TRIA(NEQ,MBAND,GK,MB)
C
```

```
  415 CONTINUE
      IF(NVIS.EQ.0) GO TO 418
      DO 416 I=1,NVIS
      K = IVIS(I)
      RLV(K) = RLV(K)+BB*VISC(I)*(BB1*VEL(K)+BB2*ACC(K))
  416 CONTINUE
  418 CONTINUE
C
      CALL BACKS(NEQ,GK,RLV,MB)
C     WRITE(6,*) "AFTER BACKS"
C
      DO 420 I=1,NEQ
      UI(I) = RLV(I)
      IF(NTER.EQ.1) GO TO 420
      DI = DISP(I)+A1*UI(I)+A2*VEL(I)+A3*ACC(I)
      VE = B1*UI(I)+B2*VEL(I)+B3*ACC(I)
      AC = C1*UI(I)+C2*VEL(I)+C3*ACC(I)
      DISP(I) = DI
      VEL(I) = VE
      ACC(I) = AC
  420 CONTINUE
      IF(NTER.EQ.1) GO TO 460
C
      IF(NTHS.EQ.0) GO TO 430
      IF(NTHS.EQ.2) GO TO 430
      CALL DVAPR(NUMNP,DISP,VEL,ACC,ID,TM,NTCSF,40)
  430 CONTINUE
C
      MMM = 1
      NNN = NNN+1
      IF(NNN.NE.NPRINT) GO TO 450
      CALL MDISC(NDC,NSG,MSTR,MDIS,MCO,1)
      MMM = 0
      NNN = 0
C
      IF(NPFL.GE.3) GO TO 450
      CALL DVAPR(NUMNP,DISP,VEL,ACC,ID,TM,NTCSF,MDIS)
  450 CONTINUE
C
      CALL EDPLV(AO,OAO,NEQ)
  460 IF(NFIRST.EQ.0) GO TO 470
      IF(NTER.EQ.0) GO TO 470
      IF(NTER.EQ.1.AND.ISAS.EQ.0) GO TO 470
      GO TO 900
  470 NDSKT = NDSK1
      NDSK1 = NDSK2
      NDSK2 = NDSKT
      NFIRST = 1
  900 IF(NCL.LE.NCYCL) GO TO 100
C
 2010 FORMAT(1H1,/,*...... STRESSES AT TIME = *,E10.3,* .....*//)
 2020 FORMAT(3X,*ELEM NO*,4X,*PNT NO*,7X,*SIGR*,7X,*SIGZ*,7X,
     .  *SIGT*,6X,*SIGRZ*,9X,*PF*/
     . 27X,*EPSR*,7X,*EPSZ*,7X,*EPST*,6X,*EPSRZ*,9X,*YF*//)
C
  950 CALL MDISC(NDC,NSG,MSTR,MDIS,MCO,2)
      RETURN
      END
C $EMA /OUTPS/,/OUTHS/
      SUBROUTINE DVAPR(NUMNP,DISP,VEL,ACC,ID,TIME,NTCSF,MDIS)
      COMMON /OUTPS/ NPFL,NDC,NSG,NPRINT,NPEL,NPMT,NPRT(1500),NPM(1500)
      COMMON /OUTHS/ NTHS,NHPEL,NHPMT,NHPRT(100),NHPM(100)
      DIMENSION ID(NUMNP,4),DISP(1),VEL(1),ACC(1),DS(4,3),DSS(2,3)
C     EMA DISP,VEL,ACC,ID
C
      IF(MDIS.EQ.40) GO TO 210
      WRITE(MDIS,2017) TIME
      IF(NTCSF.NE.2) WRITE(MDIS,2011)
      IF(NTCSF.EQ.2) WRITE(MDIS,2018)
  210 ICONT = 1
      IHCONT = 1
      DO 240 I=1,NUMNP
      IF(MDIS.NE.40) GO TO 225
      IF(IHCONT.GT.NHPMT) GO TO 240
      IF(NHPM(IHCONT).NE.I) GO TO 240
      IHCONT = IHCONT+1
      GO TO 230
  225 IF(ICONT.GT.NPMT) GO TO 240
      IF(NPM(ICONT).NE.I) GO TO 240
      ICONT = ICONT+1
  230 IF(NTCSF.NE.2) CALL SZERO(DS,12)
      IF(NTCSF.EQ.2) CALL SZERO(DSS,6)
      NN = 4
      IF(NTCSF.EQ.2) NN = 2
      DO 235 J=1,NN
      K = ID(I,J)
      IF(K) 235,235,232
  232 IF(NTCSF.NE.2) GO TO 233
      DSS(J,1) = DISP(K)
      DSS(J,2) = VEL(K)
      DSS(J,3) = ACC(K)
      GO TO 235
  233 DS(J,1) = DISP(K)
      DS(J,2) = VEL(K)
      DS(J,3) = ACC(K)
  235 CONTINUE
      IF(NTCSF.NE.2) WRITE(MDIS,2012) I,DS
      IF(NTCSF.EQ.2) WRITE(MDIS,2019) I,DSS
  240 CONTINUE
C
 2011 FORMAT(
     .  4X,*NODE*,17X,*DISPLACEMENT*,30X,*VELOCITY*,30X,*ACCELERATION*/
     .4X,*NUMBER*,5X,*UR*,8X,*UZ*,8X,*WR*,8X,*WZ*,8X,*UR*,8X,*UZ*,8X,
     .  *WR*,8X,*WZ*,8X,*UR*,8X,*UZ*,8X,*WR*,8X,*WZ*//)
 2012 FORMAT(I10,12E10.3)
 2017 FORMAT(1H1,*...... RESPONSE AT TIME = *,E10.3,*  ......*//)
 2018 FORMAT(4X,*NODE*,6X,*DISPLACEMENT*,10X,*VELOCITY*,10X,
     . *ACCELERATION*,/4X,*NUMBER*,5X,*UR*,8X,*UZ*,8X,*UR*,8X,
     . *UZ*,8X,*UR*,8X,*UZ*//)
 2019 FORMAT(I10,6E10.3)
      RETURN
      END
      SUBROUTINE STRSP(NEL,NIP,S,P,E,KS,KF,NTCSF,YF,MSTR)
      DIMENSION S(4,16),P(1),E(4,16),YF(16)
C     EMA P
      WRITE(MSTR,2010) NEL
      IF(KF.EQ.1.OR.NTCSF.EQ.2) CALL EZERO(P,NIP)
      DO 100 I=1,NIP
      WRITE(MSTR,2020) (I,(S(J,I),J=1,4),P(I))
      WRITE(MSTR,2022) (  (E(J,I),J=1,4),YF(I))
  100 CONTINUE
      WRITE(MSTR,2030)
 2010 FORMAT(I10)
 2020 FORMAT(15X,I5,5E11.4)
 2022 FORMAT(20X,4E11.4,F11.2)
 2030 FORMAT(//)
      RETURN
      END
      SUBROUTINE STRCL(MSTR,NUM,NMPK,ESTRN,MXIP,MXSH,NLNR,NTCSF,NINT,
     *  KS,KF,SS,PSA,PFA)
      DIMENSION SS(4,16),PSA(MXSH,1),PFA(1),ESTRN(4,16),YF(16)
C     EMA NINT,PSA,PFA
C
      DO 310 I=1,MXIP
      YF(I) = PSA(7,I)
      IF(NLNR.EQ.1) YF(I) = PSA(8,I)
      DO 310 J=1,4
  310 SS(J,I) = PSA(J,I)
C
      IF(NMPK.NE.0) CALL SIGN(SS,MXIP*4)
      CALL STRSP(NUM,MXIP,SS,PFA,ESTRN,KS,KF,NTCSF,YF,MSTR)
C
      RETURN
      END
      SUBROUTINE DLOAD(TS,GACL,ID,XM,AO,TM,DYL,IEQH,CINT,TD,XC,
     *   UNV,MBM,LSTRT,NUMNP,NEQ,NUMLH,C2,C3)
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      COMMON /GCALP/ NCYCL,DT,NUPDAT,ITER
      COMMON /MMBDP/ NUMNX,NUMEL,NUMMAT,NVIS,NSKEW,NEQX,MBAND
      COMMON /ACCLF/ MTYPE,NUMAP,DTXA
      COMMON /PRELF/ NUMLP,NUMLX,NUMTP,NTYPE,DTXX
      DIMENSION TS(1),GACL(1),ID(NUMNP,1),XM(NEQ,1),AO(1),UNV(1),MBM(1),
     *   DYL(NUMLH,1),IEQH(3,1),CINT(1),TD(1),XC(1)
```

```
C     EMA TS,GACL,ID,XM,AO,DYL,IEQH,CINT,TD,XC,UNV,MBM
      IF(NFG.NE.1) GO TO 300
      IF(LSTRT.EQ.1) GO TO 150
C
C     BASE ACCELERATION ( NFG = 1 )
C      SET UP LOAD MATRIX
C
      L = 2
      DTX = DTXA
      IF(MTYPE.NE.0) DTX = TS(2)-TS(1)
      NC = 0
      CALL EZERO(AO,NEQ)
C
  150 CONTINUE
      IF(MTYPE.NE.0) GO TO 154
C
      NC = NC+1
      ACCL = GACL(NC+1)
      GO TO 161
C
  154 CONTINUE
      IF(TS(L)-TM) 155,155,160
  155 DTX = TS(L+1)-TS(L)
      L = L+1
      IF(DTX) 158,155,160
  158 WRITE(6,2010) TS(L-1)
      STOP
C
  160 SLOPE = (GACL(L)-GACL(L-1))/DTX
      ACCL = GACL(L-1)+(TM-TS(L-1))*SLOPE
  161 CONTINUE
C
      ACCL = ACCL/C2
C
      IF(ICONST.NE.0) GO TO 182
C
      DO 180 I=1,NUMNP
      DO 170 J=1,2
      II = J+J-1
      JJ = ID(I,II)
      IF(JJ) 170,170,165
  165 AO(JJ) = -ACCL*XM(JJ,1)
  170 CONTINUE
  180 CONTINUE
C
      GO TO 195
  182 CONTINUE
      CALL MLTPL(XM,AO,UNV,MBM,NEQ)
C
      DO 185 I=1,NEQ
  185 AO(I) = -ACCL*AO(I)
```

```
C
  195 LSTRT = 1
C
      RETURN
C
  300 CONTINUE
C
      K = 1
C
C      PRESSURE LOAD ( NFG = 2 OR 4 )
C
  500 CONTINUE
      IF(TM.GE.TD(K).AND.TM.LE.TD(K+1)) GO TO 520
      IF(TM.LT.TD(K)) K=K-1
      IF(TM.GT.TD(K+1)) K=K+1
      GO TO 500
  520 TD1 = TD(K)
      DTXX = TD(K+1)-TD(K)
C
      DO 700 I=1,NUMLP
      JJ = IEQH(1,I)
      IF(JJ.LE.0) GO TO 700
      LHNO = IEQH(2,I)
      C = CINT(I)
      SLOPE = C*(DYL(LHNO,K+1)-DYL(LHNO,K))/DTXX
      AO(JJ) = C*DYL(LHNO,K)+(TM-TD1)*SLOPE
      IF(IP.NE.4) GO TO 700
      AO(JJ) = 0.5*AO(JJ)
  700 CONTINUE
      LSTRT = 1
      RETURN
 2010 FORMAT(/*ERROR IN BASE ACCEL OR DYN LOAD, TIME..... = *,E10.3)
C
      END
      SUBROUTINE VXDUP(A,B)
      DIMENSION A(4),B(4)
C     EMA A
      DO 100 I=1,4
  100 A(I) = B(I)
      RETURN
      END
      SUBROUTINE DUPVT(A,B)
      DIMENSION A(4),B(4)
      DO 100 I=1,4
      B(I)=A(I)
100   CONTINUE
      RETURN
      END
```

```
C FTN7X,L
C $EMA /NALLO/
      SUBROUTINE MDISC(NDC,NSG,MSTR,MDIS,MCO,NF)
C     COMMON /NALLO/ NSDFL(8,9,2,2)
      INTEGER DINO(2,2),SDPR(2,2),BLOC(2,2),CM,CC,MM(2),NAME(8),NO(9)
C     EMA NDC,NSG
C     DATA DINO /'17193233'/
C     DATA SDPR /'STPRDIPR'/
C     DATA BLOC /' 100  50'/
C     DATA NO /'1:2:3:4:5:6:7:8:9:'/
C     DATA CM /':-'/
C     DATA CC /'::'/
C
      MCO = MCO+1
      IF(NF.EQ.2) GO TO 600
      IF(NSG.NE.0) GO TO 400
C
C     NSG = 0   WRITE OUTPUT IN ONE FILE
C
      IF(MCO.NE.1) RETURN
      IF(NDC.NE.0) GO TO 200
C
C     NDC = 0   WRITE OUTPUT ON HARD DISC
C
      MSTR = 6
      MDIS = 6
      RETURN
C
C     NDC = 1 WRITE OUTPUT ON FLOPPY DISC
C
  200 CONTINUE
      MSTR = 7
      MDIS = 7
      RETURN
C
C     NSG = 1   WRITE OUTPUT IN MULTI-SEGMENTED FILES
C
  400 CONTINUE
      IF(MCO.NE.1) GO TO 500
C
C     MCO = 1  INITIALIZE DISC NAMES
C
      II = NDC+1
C     DO 460 I=1,2
C     DO 440 J=1,9
C     NSDFL(1,J,I,II) = SDPR(1,I)
C     NSDFL(2,J,I,II) = SDPR(2,I)
C     NSDFL(3,J,I,II) = NO(J)
C     NSDFL(4,J,I,II) = CM
C     NSDFL(5,J,I,II) = DINO(I,II)
```

```
C     NSDFL(6,J,I,II) = CC
C     NSDFL(7,J,I,II) = BLOC(1,I)
C     NSDFL(8,J,I,II) = BLOC(2,I)
C 440 CONTINUE
C 460 CONTINUE
      MSTR = 60
      MDIS = 70
  500 CONTINUE
      MSTR = MSTR+1
      MDIS = MDIS+1
      MM(1) = MSTR
      MM(2) = MDIS
      II = NDC+1
C     IF(MCO.NE.1) CLOSE(MSTR-1)
C     IF(MCO.NE.1) CLOSE(MDIS-1)
C     DO 560 I=1,2
C     DO 540 J=1,8
C 540 NAME(J) = NSDFL(J,MCO,I,II)
C     OPEN(MM(I),FILE=NAME,STATUS=*NEW*)
C 560 CONTINUE
      RETURN
  600 CONTINUE
C
C     CLOSE THE OPENED FILES
C
      IF(NSG.EQ.0) RETURN
C     CLOSE (MSTR)
C     CLOSE (MDIS)
      RETURN
      END
C     SUBROUTINE ALPHA(EST)
C     COMMON /INTCN/ AA(18),ALPA
C     DIMENSION EST(16,16)
C     FAC = 1.0+ALPA
C     DO 100 I=1,16
C     DO 100 J=1,16
C 100 EST(I,J) = FAC*EST(I,J)
C     RETURN
C     END
      SUBROUTINE ESMBL(ST,VC,EST,EVC,NDR,NDC,NEQ,MBAND,NDOF,
     *   NF,NA,NAC)
C
      DIMENSION ST(NEQ,MBAND),VC(NEQ),EST(NDOF,1),EVC(1),
     *   NDR(1),NDC(1)
C     EMA ST,VC
C
C     GLOBAL MATRIX AND VECTOR ASSEMBLY
C       NF = 1 MATRIX AND VECTOR
C       NF = 2 MATRIX ONLY
C       NF = 3 VECTOR ONLY
C
```

```
      IF(NF.EQ.3) GO TO 250
      DO 200 K=1,NA
      IR = NDR(K)
      IF(IR.LE.0) GO TO 200
      DO 100 L=1,NAC
      IC = NDC(L)
      IF(IC.LE.0) GO TO 100
      IC = IC-IR+1
      IF(IC.LE.0) GO TO 100
      ST(IR,IC) = ST(IR,IC)+EST(K,L)
  100 CONTINUE
  200 CONTINUE
      IF(NF.EQ.2) GO TO 350
  250 CONTINUE
      DO 300 K=1,NA
      IR = NDR(K)
      IF(IR.LE.0) GO TO 300
      VC(IR) = VC(IR)+EVC(K)
  300 CONTINUE
  350 CONTINUE
      RETURN
      END
      SUBROUTINE TRIA(NEQ,M,A,MB)
      DIMENSION A(NEQ,1),MB(NEQ)
C     EMA A,MB
C
C     WRITE(1,*) *BEFORE TRIA*
      NE = NEQ-1
      MN = M-1
      MM = M
      MK = NEQ-MN
      DO 300 N=1,NE
      NT = N-MK
      IF(NT.GT.0) MM = MM-1
      MB(N) = 0
      IF(A(N,1).EQ.0.0) GO TO 300
      L = N
      IH = MM
      JB = 0
      IB = 0
      DO 200 I=2,IH
      L = L+1
      J = L
      IB = IB+1
      AI = A(N,I)
      C = AI/A(N,1)
      IF(C.EQ.0.0) GO TO 200
      JC = 1
      DO 100 K=I,IH
      A(J,JC) = A(J,JC)-C*A(N,K)
  100 JC = JC+1
```

```
      A(N,I) = C
      JB = IB
  200 CONTINUE
      MB(N) = JB
  300 CONTINUE
      MB(NEQ) = 0
      RETURN
      END
      SUBROUTINE BACKS(NN,A,B,MB)
      DIMENSION A(NN,1),B(NN),MB(NN)
C     EMA A,B,MB
C     WRITE(1,*) *BEFORE BACKS*
      N = 0
  270 N = N+1
      C = B(N)
      IF(A(N,1).NE.0.0) B(N) = B(N)/A(N,1)
      IF(N.EQ.NN) GO TO 300
      IL = N+1
      IH = N+MB(N)
      M = 1
      DO 285 I=IL,IH
      M = M+1
  285 B(I) = B(I)-A(N,M)*C
      GO TO 270
C
  300 IL = N
      N = N-1
      IF(N.EQ.0) RETURN
      IH = N+MB(N)
      M = 1
      C = B(N)
      DO 400 I=IL,IH
      M = M+1
  400 C = C-A(N,M)*B(I)
      B(N) = C
      GO TO 300
      END
      SUBROUTINE MLTPL(A,B,BO,MB,NEQ)
      DIMENSION A(NEQ,1),B(1),BO(1),MB(1)
C     EMA A,B,BO,MB
      DO 300 N=1,NEQ
      BB = A(N,1)*BO(N)
      L = N
      NI = MB(N)
      IF(NI.GT.1) GO TO 50
      GO TO 120
   50 DO 100 M=2,NI
      L = L+1
  100 BB = BB+A(N,M)*BO(L)
  120 L = N
      NJ = MB(N+NEQ)
      IF(NJ.GT.1) GO TO 150
      GO TO 250
  150 DO 200 M=2,NJ
      L = L-1
  200 BB = BB+A(N-M+1,M)*BO(L)
  250 B(N) = B(N)+BB
  300 CONTINUE
      RETURN
      END
      SUBROUTINE TDPLV(A,B,NR,NC)
      DIMENSION A(NR,NC),B(NR,NC)
C     EMA A,B
      DO 100 J=1,NC
      DO 100 I=1,NR
      AA = A(I,J)
  100 B(I,J) = AA
      RETURN
      END
      SUBROUTINE EDPLV(A,B,N)
      DIMENSION A(N),B(N)
C     EMA A,B
      DO 100 I=1,N
  100 B(I) = A(I)
      RETURN
      END
C     SUBROUTINE DOT(X,Y,FF,N)
C     DIMENSION X(N),Y(N)
C     FF = 0.0
C     DO 100 I=1,N
C 100 FF = FF+X(I)*Y(I)
C     RETURN
C     END
      SUBROUTINE SZERO(A,N)
      DIMENSION A(N)
      DO 100 I=1,N
  100 A(I) = 0.0
      RETURN
      END
      SUBROUTINE IZERO(L,N)
      DIMENSION L(N)
      DO 100 I=1,N
  100 L(I) = 0
      RETURN
      END
      SUBROUTINE TZERO(A,NR,NC)
      DIMENSION A(NR,NC)
C     EMA A
      DO 100 I=1,NR
      DO 100 J=1,NC
  100 A(I,J) = 0.0
      RETURN
      END
      SUBROUTINE EZERO(A,N)
      DIMENSION A(N)
C     EMA A
      DO 100 I=1,N
  100 A(I) = 0.0
      RETURN
      END
      SUBROUTINE ELSTF(B,D,EN,XBAR,FAC,NC,NCC,S,IP,NSYM)
      DIMENSION B(2,NC),D(4,4),EN(NC),S(16,NCC)
C     EMA B,EN,XBAR,FAC
C
      IF(IP.GT.2) GO TO 250
      DO 200 J=1,NC
      B1 = B(1,J)
      B2 = B(2,J)
      DB1 = D(1,1)*B1+D(1,4)*B2
      DB2 = D(2,1)*B1+D(2,4)*B2
      DB4 = D(4,1)*B1+D(4,4)*B2
C
      DBB1 = D(1,2)*B2+D(1,4)*B1
      DBB2 = D(2,2)*B2+D(2,4)*B1
      DBB4 = D(4,2)*B2+D(4,4)*B1
C
      DO 100 I=1,J
      S(I,J) = S(I,J)+(B(1,I)*DB1+B(2,I)*DB4)*FAC
      S(I+NC,J+NC) = S(I+NC,J+NC)+(B(2,I)*DBB2+B(1,I)*DBB4)*FAC
      S(J,I) = S(I,J)
      S(J+NC,I+NC) = S(I+NC,J+NC)
  100 CONTINUE
C
      DO 150 I=1,NC
      S(I,J+NC) = S(I,J+NC)+(B(1,I)*DBB1+B(2,I)*DBB4)*FAC
      S(J+NC,I) = S(I,J+NC)
  150 CONTINUE
  200 CONTINUE
      GO TO 500
C
  250 CONTINUE
      IF(IP.NE.3) GO TO 410
C
      DO 400 J=1,NC
      B1 = B(1,J)
      B2 = B(2,J)
C
      DB1 = D(1,1)*B1+D(1,4)*B2+D(1,3)*EN(J)/XBAR
      DB2 = D(2,1)*B1+D(2,4)*B2+D(2,3)*EN(J)/XBAR
      DB3 = D(3,1)*B1+D(3,4)*B2+D(3,3)*EN(J)/XBAR
      DB4 = D(4,1)*B1+D(4,4)*B2+D(4,3)*EN(J)/XBAR
C
      DBB1 = D(1,2)*B2+D(1,4)*B1
      DBB2 = D(2,2)*B2+D(2,4)*B1
      DBB3 = D(3,2)*B2+D(3,4)*B1
      DBB4 = D(4,2)*B2+D(4,4)*B1
C
      DO 300 I=1,J
      S(I,J) = S(I,J)+(B(1,I)*DB1+B(2,I)*DB4+DB3*EN(I)/XBAR)*FAC
      S(I+NC,J+NC) = S(I+NC,J+NC)+(B(2,I)*DBB2+B(1,I)*DBB4)*FAC
      S(J,I) = S(I,J)
      S(J+NC,I+NC) = S(I+NC,J+NC)
  300 CONTINUE
C
      DO 350 I=1,NC
      S(I,J+NC) = S(I,J+NC)+(B(1,I)*DBB1+B(2,I)*DBB4+DBB3*EN(I)/XBAR)*FAC
      S(J+NC,I) = S(I,J+NC)
  350 CONTINUE
C
  400 CONTINUE
      GO TO 500
  410 CONTINUE
      IF(IP.NE.4) GO TO 500
      DO 480 J=1,NC
      B1 = B(1,J)
C
      DB1 = B1*D(1,1)+(D(1,2)+D(1,3))*EN(J)/XBAR
      DB2 = B1*D(2,1)+(D(2,2)+D(2,3))*EN(J)/XBAR
      DB3 = B1*D(3,1)+(D(3,2)+D(3,3))*EN(J)/XBAR
C
      DO 420 I=1,J
      S(I,J) = S(I,J)+(B(1,I)*DB1+(DB2+DB3)*EN(I)/XBAR)*FAC
      S(J,I) = S(I,J)
  420 CONTINUE
C
  480 CONTINUE
C
  500 CONTINUE
      RETURN
      END
      SUBROUTINE ELSTN(B,EN,XBAR,DU,STRN,NC,IP)
      DIMENSION B(2,NC),EN(NC),DU(1),STRN(4)
C     EMA B,EN,XBAR
C
      S1 = 0.0
      S2 = 0.0
      S3 = 0.0
      S4 = 0.0
C
      DO 10 L=1,NC
      S1 = S1+B(1,L)*DU(L)
      S2 = S2+B(2,L)*DU(L+NC)
      IF(IP.GT.2) GO TO 3
      S3 = 0.0
      GO TO 5
    3 S3 = S3+EN(L)*DU(L)/XBAR
    5 S4 = S4+B(2,L)*DU(L)+B(1,L)*DU(L+NC)
   10 CONTINUE
C
      STRN(1) = S1
      STRN(2) = S2
      STRN(3) = S3
      STRN(4) = S4
      IF(IP.NE.4) GO TO 20
      STRN(2) = STRN(3)
      STRN(4) = 0.0
   20 CONTINUE
C
      RETURN
      END
      SUBROUTINE ELAST(B,G,C)
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      DIMENSION C(4,4)
C
C     COMPUTE ELASTIC  C - MATRIX
C
      IF(IP.GT.1) GO TO 20
      DNU = (3.*B-2.*G)/(6.*B+2.*G)
      A = 2.0*G/(1.0-DNU)
      D = A*DNU
      GO TO 50
   20 CONTINUE
C
      A = B+4.*G/3.
      D = A-2.*G
C
   50 CONTINUE
      DO 100 I=1,3
      C(I,4) = 0.
      C(4,I) = 0.
      DO 100 J=1,3
      IF(I.EQ.J) C(I,J) = A
      IF(I.NE.J) C(I,J) = D
  100 CONTINUE
      C(4,4) = G
C
      IF(IP.GT.1) GO TO 120
      C(1,3) = 0.0
      C(2,3) = 0.0
      C(3,3) = 0.0
      C(3,1) = 0.0
      C(3,2) = 0.0
  120 CONTINUE
      RETURN
      END
      SUBROUTINE RESLS(B,S,EN,XBAR,WIWJ,NC,RS,IP)
      DIMENSION B(2,1),S(1),EN(1),RS(1)
C     EMA B,EN,XBAR,WIWJ
C
C     SOLID PHASE ELEMENT LOAD VECTOR DUE TO EFFECTIVE STRESS
C
      DO 100 I=1,NC
      RS(I) = RS(I)-WIWJ*(B(1,I)*S(1)+B(2,I)*S(4))
      IF(IP.EQ.3) RS(I) = RS(I)-WIWJ*S(3)*EN(I)/XBAR
      IF(IP.EQ.4) RS(I) = RS(I)-2.*WIWJ*S(3)*EN(I)/XBAR
      RS(I+NC) = RS(I+NC)-WIWJ*(B(2,I)*S(2)+B(1,I)*S(4))
  100 CONTINUE
      RETURN
      END
      SUBROUTINE EXTRT(DU,NA,GU,N)
      DIMENSION DU(1),NA(1),GU(1)
C     EMA GU
      CALL SZERO(DU,N)
      DO 100 I=1,N
      J = NA(I)
      IF(J.LE.0) GO TO 100
      DU(I) = GU(J)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE SIGN(SS,N)
      DIMENSION SS(N)
C
C     CHANGE THE SIGN OF STRESSES OR STRAINS
C
      DO 100 I=1,N
  100 SS(I) = -SS(I)
      RETURN
      END
      SUBROUTINE WREAD(A,N,B,NF)
      DIMENSION A(N),B(N)
C     EMA A,B
      DO 100 I=1,N
      IF(NF.EQ.0) B(I) = A(I)
      IF(NF.NE.0) A(I) = B(I)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE WRID3(A,N,B,NF)
      DIMENSION A(N),B(N)
C     EMA A
      DO 100 I=1,N
      IF(NF.EQ.0) B(I) = A(I)
      IF(NF.NE.0) A(I) = B(I)
  100 CONTINUE
      RETURN
      END
C
      SUBROUTINE TRANS(NTP,A,N,NE,B)
      DIMENSION A(NE,N),B(NE)
C     EMA A
      REWIND NTP
      DO 200 I=1,N
      READ (NTP) B
      DO 100 J=1,NE
  100 A(J,I) = B(J)
  200 CONTINUE
      REWIND NTP
      RETURN
      END
C     SUBROUTINE TAUMX(S,SMAX)
C     DIMENSION S(4)
C     SMAX = SQRT((0.5*(S(1)-S(2)))**2+S(4)*S(4))
C     RETURN
C     END
      SUBROUTINE VELBC(NLNR,NFIRST,IVEL,NUMVH,NUMVTP,NEQ,MBAND,ID,GK,
     * GKO,RLV,VEL,ACC,KVEL,KEQH,VINT,TV,SVEL,TM,NUMNP,NVTYPE,DTXV)
      COMMON /INTCN/ AA(9),B1,B2,B3,BB(7)
      DIMENSION ID(NUMNP,1),GK(NEQ,1),GKO(NEQ,1),RLV(1),VEL(1),ACC(1),
     *  KVEL(1),KEQH(1),VINT(1),TV(1),SVEL(NUMVTP,1)
C     EMA ID,GK,GKO,RLV,VEL,ACC,KVEL,KEQH,VINT,TV,SVEL
C
C     MODIFY LOAD VECTOR AND GLOBAL STIFFNESS MATRIX
C
      K = 1
      DO 700 N=1,IVEL
      LHNO = KEQH(N)
      KK = KVEL(N)
      CALL LIFNT(TM,NUMVTP,TV,SVEL(1,LHNO),SVM,K)
      SVM = SVM*VINT(N)
      DLTAU = (SVM-B2*VEL(KK)-B3*ACC(KK))/B1
      IF(NLNR.EQ.0.AND.NFIRST.NE.0) GO TO 450
      CALL MGKRL(GK,RLV,KK,DLTAU,NEQ,MBAND,0,IVEL,KVEL)
      GO TO 700
  450 CALL MGKRL(GKO,RLV,KK,DLTAU,NEQ,MBAND,1,IVEL,KVEL)
  700 CONTINUE
      RETURN
      END
      SUBROUTINE LIFNT(TM,NUMVTP,TV,SV,SVM,K)
      DIMENSION TV(1),SV(1)
C     EMA TV,SV
  100 IF(TM.GE.TV(K).AND.TM.LE.TV(K+1)) GO TO 200
      IF(TM.GT.TV(K+1)) K = K+1
      IF(TM.LT.TV(K)) K = K-1
      IF(K.LT.NUMVTP) GO TO 100
      WRITE(6,*) " ERROR IN VELOCITY TIME POINTS "
      STOP
  200 CONTINUE
      SVM = SV(K)+(SV(K+1)-SV(K))*(TM-TV(K))/(TV(K+1)-TV(K))
      RETURN
      END
      SUBROUTINE MGKRL(GK,RLV,N,DLTAU,NEQ,MBAND,NF,IVEL,KVEL)
      DIMENSION GK(NEQ,MBAND),RLV(NEQ),KVEL(1)
C     EMA GK,RLV,KVEL
      DO 400 M=2,MBAND
      K = N-M+1
      IF(K.LE.0) GO TO 100
      CALL MDRLV(IVEL,KVEL,K,RLV(K),GK(K,M),DLTAU)
      IF(NF.EQ.0) GK(K,M) = 0.0
  100 L = N+M-1
      IF(NEQ-L) 300,200,200
  200 CALL MDRLV(IVEL,KVEL,L,RLV(L),GK(N,M),DLTAU)
  300 IF(NF.EQ.0) GK(N,M) = 0.0
  400 CONTINUE
      IF(NF.EQ.0) GK(N,1) = 1.0
      RLV(N) = DLTAU
      RETURN
      END
      SUBROUTINE MDRLV(IVEL,KVEL,K,R,G,D)
      DIMENSION KVEL(1)
C     EMA KVEL,R,G
      NCK = 0
      DO 100 I=1,IVEL
  100 IF(KVEL(I).EQ.K) NCK = 1
      IF(NCK.EQ.0) R = R-G*D
      RETURN
      END
      SUBROUTINE MSKEW(E,NDOF,NC,NCC,I,J,K,IROT,C,S)
      DIMENSION E(NDOF,1)
      IF(IROT.EQ.2) GO TO 150
C
      DO 100 L=1,NCC
      ELI = E(L,I)
      ELINC = E(L,I+NC)
      ELJ = E(L,J)
      ELJNC = E(L,J+NC)
      E(L,I) = ELI*C+ELINC*S
      E(L,I+NC) = ELI*(-S)+ELINC*C
      E(L,J) = ELJ*C+ELJNC*S
      E(L,J+NC) = ELJ*(-S)+ELJNC*C
      IF(K.EQ.0) GO TO 100
      ELK = E(L,K)
      ELKNC = E(L,K+NC)
      E(L,K) = ELK*C+ELKNC*S
      E(L,K+NC) = ELK*(-S)+ELKNC*C
  100 CONTINUE
C
      IF(IROT.EQ.1) RETURN
  150 CONTINUE
C
      DO 200 L=1,NCC
      EIL = E(I,L)
      EINCL = E(I+NC,L)
      EJL = E(J,L)
      EJNCL = E(J+NC,L)
      E(I,L) = EIL*C+EINCL*S
      E(I+NC,L) = EIL*(-S)+EINCL*C
      E(J,L) = EJL*C+EJNCL*S
      E(J+NC,L) = EJL*(-S)+EJNCL*C
      IF(K.EQ.0) GO TO 200
      EKL = E(K,L)
      EKNCL = E(K+NC,L)
      E(K,L) = EKL*C+EKNCL*S
      E(K+NC,L) = EKL*(-S)+EKNCL*C
  200 CONTINUE
      RETURN
      END
      SUBROUTINE DUPVX(A,B)
      DIMENSION A(4),B(4)
C     EMA B
      DO 100 I=1,4
  100 A(I) = B(I)
      RETURN
      END
      SUBROUTINE MTPKO(STRS,STRN,BK,G)
      COMMON /GENOP/ NF,NTCSF,ISFG,IP,NLNR,ICONST,NFG
      DIMENSION STRS(4),STRN(4)
C     EMA STRS
      IF(IP.GT.1) GO TO 20
      DNU = (3.*BK-2.*G)/(6.*BK+2.*G)
      A = 2.*G/(1.-DNU)
      B = A*DNU
      GO TO 50
   20 CONTINUE
      A = BK+4.*G/3.
      B = A-2.*G
   50 CONTINUE
      STRS(1) = STRS(1)+A*STRN(1)+B*(STRN(2)+STRN(3))
      STRS(2) = STRS(2)+A*STRN(2)+B*(STRN(1)+STRN(3))
      IF(IP.EQ.1) STRS(3) = 0.0
      IF(IP.NE.1) STRS(3) = STRS(3)+A*STRN(3)+B*(STRN(1)+STRN(2))
      STRS(4) = STRS(4)+G*STRN(4)
      RETURN
      END
```

```
C FTN7X,L
C $EMA /MDCPL/
      SUBROUTINE MTPK1(PS,DSTRN,DP,NFIRST,NFSTF,NMAT,BKE,GE,PROP)
      COMMON/MDCPL/ BLN(11,2,5),BUN(11,2,5),TYF(11,3,5),NLP(5),
     *              NUP(5),NTP(5),NCO(5)
      DIMENSION DSTRN(4),DP(4,4),PS(1),PROP(1)
C     EMA PS,PROP
      GP=PROP(8)
      A=PROP(9)
      B=PROP(10)
      IF(NFIRST.NE.0) GO TO 300
C
C     NFIRST = 0 ,  ASSUME ELASTIC
C
      PMAX=(PS(1)+PS(2)+PS(3))/3.
      YPMAX=0.0
      PS(7)=PMAX
      PS(8)=YPMAX
      BK=BKE
      G=GE
      GO TO 800
C
C     NFIRST > 0
C
300   CONTINUE
      N=NMAT
      CALL DCOUP(PS(1),DSTRN,BLN(1,1,N),NLP(N),BUN(1,1,N),NUP(N),
     *           TYF(1,1,N),NTP(N),PS(7),PS(8),A,B,GE,GP,BK,G)
C
C     CALCULATE STRESS-STRAIN MATRIX
C
800   CONTINUE
      IF(NFSTF.EQ.0) RETURN
      CALL ELAST(BK,G,DP)
      RETURN
      END
C
      SUBROUTINE DCOUP(STRS,DSTRN,BLN,NLP,BUN,NUP,TYF,NTP,PMAX,
     *                 YPMAX,A,B,GE,GP,BK,G)
C
      DIMENSION STRS(4),DSTRN(4),BLN(11,2),BUN(11,2),TYF(11,3),
     *          DDSRN(4),DDVSR(4),DST(4),DVSR(4),DSIC(4),DVSN(4)
C     EMA STRS,BLN,NLP,BUN,NUP,TYF,NTP,PMAX,YPMAX
C
C
C     INPUT  :
C             STRS(4):         (OLD STRESS)
C             DSTRN(4):        (CURRENT STRAIN INCREMENT)
C             BLN(NLP,2):      P-BLK  CURVE
C             BUN(NUP,2):      P-BUK  CURVE
```

```
C             TYF(NTP,3):       P-TY-TF  CURVE
C             PMAX:             PREVIOUS MAX. MEAN PRESSURE
C             YPMAX:             PREVIOUS MAX. Y'
C             GE:               ELASTIC SHEAR MODULUS
C             GP:               PLASTIC SHERA MODULUS
C             A:                RATIO OF INITIAL STRENGTH
C                              OO YIELD STRENGTH
C                               ENVELOPE
C             B:                FRACTION OF PLASTIC SHEAR MODULUS
C                               AT FAILURE
C
C
C     OUTPUT  :
C              STRS(4):             (UPDATE STRESS)
C              PMAX:            UPDATE PMAX
C              YPMAX:           UPDATE Y'MAX
C              BK:              UPDATE BULK MODULUS
C              G:               UPDATE SHEAR MODULUS
C
C     SET THE NUMBER OF STRAIN SUBINCREMENTS INTO 10
C
      NDIV=10
      DO 10 I=1,4
      DDSRN(I)=DSTRN(I)/NDIV
10    CONTINUE
C
C
      DO 1000 II=1,NDIV
C
C     DECOMPOSE INTO DEVIATORIC AND VOLUMETRIC PARTS
C
      CALL DECOM(STRS,DDSRN,PP,PDVSR,DV,DVSN)
C
C     COMPUTE BULK MODULUS(BK)
C
      IF(PP.GE.PMAX) CALL MDLUS(BLN(1,1),BLN(1,2),NLP,PP,BK)
      IF(PP.LT.PMAX) CALL MDLUS(BUN(1,1),BUN(1,2),NUP,PP,BK)
C
C     UPDATE MEAN PRESSURE AND PMAX
C
      P=PP+BK*DV
      IF(P.GT.PMAX) PMAX=P
C
C     COMPUTE SHEAR MODULUS(G)
C
      IF(YPMAX.GE.A) GO TO 200
C
C     Y'MAX < A , IN ELASTIC STATE
C
      G=GE
      GO TO 800
```

```
C
C     Y'MAX >= A , IN PLASTIC STATE
C
200   CONTINUE
C
C     CALCULATEE Y' BASED ON OLD STRESS STATE
C
      CALL CALYF(TYF(1,1),TYF(1,2),TYF(1,3),PP,TAUY,TAUF,NTP)
      CALL TAUOC(PDVSR,TAU)
      CALL CALYP(TAU,TAUY,TAUF,YPMAX,YP)
C
      IF(YP.GE.YPMAX) GO TO 400
C
C     Y' < Y'MAX , UNLOADING
      G=GE
      GO TO 800
C
C     Y' >= Y'MAX, LOADING
C
400   CONTINUE
C
C     FIND TRIAL SHEAR MODULUS
C
      CALL CALGT(YP,A,B,GE,GP,GT)
C
C     CALCULATE OCTAHEDRAL SHEAR STRESS BASED ON G TRIAL
C
      CALL DEVST(DVSN,GT,DSIC)
      CALL ADDVT(PDVSR,DSIC,DST)
      CALL TAUOC(DST,TAU)
C
C     CALCULATE Y' CURRENT
C
      CALL CALYF(TYF(1,1),TYF(1,2),TYF(1,3),P,TAUY,TAUF,NTP)
      CALL CALYP(TAU,TAUY,TAUF,YPMAX,YPCUR)
C
C     CHECK WHETHER IT IS LOADING OR NOT
C
      IF(YPCUR.GE.YPMAX)GOTO 600
C
C     Y' <  Y'MAX,  UNLOADING
C
      G=GE
      GOTO 800
C
C     Y'CUR. >= Y'MAX,  LOADING
C
600   CONTINUE
      G=GT
      YPMAX=YPCUR
      CALL DUPVT(DST,DVSR)
      GO TO 900
C
C     UPDATE SHEAR STRESS AND Y'MAX
C
800   CONTINUE
      CALL DEVST(DVSN,G,DSIC)
      CALL ADDVT(PDVSR,DSIC,DVSR)
C
      CALL TAUOC(DVSR,TAU)
      CALL CALYF(TYF(1,1),TYF(1,2),TYF(1,3),P,TAUY,TAUF,NTP)
      CALL CALYP(TAU,TAUY,TAUF,YPMAX,YPCUR)
      IF(YPCUR.GT.YPMAX) YPMAX=YPCUR
C
C     UPDATE STRESSES
C
900   CONTINUE
      DO 950 I=1,3
      STRS(I)=P+DVSR(I)
950   CONTINUE
      STRS(4)=DVSR(4)
C
C
1000  CONTINUE
      RETURN
      END
C
      SUBROUTINE DECOM(STRS,DSTRN,P,DVSR,DV,DVSN)
      DIMENSION STRS(4),DSTRN(4),DVSR(4),DVSN(4)
C     EMA STRS
      P=(STRS(1)+STRS(2)+STRS(3))/3.
      DV = DSTRN(1)+DSTRN(2)+DSTRN(3)
      DO 100 I=1,3
      DVSR(I)=STRS(I)-P
      DVSN(I)=DSTRN(I)-DV/3.
100   CONTINUE
      DVSR(4)=STRS(4)
      DVSN(4)=DSTRN(4)
      RETURN
      END
C
      SUBROUTINE MDLUS(P,B,NBP,PC,BK)
      DIMENSION P(1),B(1)
C     EMA P,B,NBP
      K=1
100   IF(PC.GE.P(K).AND.PC.LE.P(K+1)) GO TO 200
      IF(PC.GT.P(K+1)) K=K+1
      IF(PC.LT.P(K)) K=K-1
      IF(K.LT.NBP) GO TO 100
      WRITE(6,1)
1     FORMAT(22H K IS GREATER THAN NBP)
      STOP
```

```
200   BK=B(K)
      RETURN
      END
C
      SUBROUTINE CALGT(YP,A,B,GE,GP,GT)
      GA= 0.5*(GE+GP)
      GD= 0.5*(GE-GP)
      IF(YP.LT.A) GT=GE
      IF(YP.GE.A.AND.YP.LT.1.0) GT=GA-GD*(2.*YP-(1.+A))/(1.-A)
      IF(YP.GE.1.0.AND.YP.LT.2.0) GT=GP
      IF(YP.GE.2.0) GT=B*GP
      RETURN
      END
C
      SUBROUTINE CALYP(TAU,TAUY,TAUF,YPMAX,YP)
C     EMA YPMAX
      IF(YPMAX.LT.1.0) YP=TAU/TAUY
      IF(YPMAX.GE.1.0.AND.YPMAX.LT.2.0)
     *  YP = 1.0+(TAU-TAUY)/(TAUF-TAUY)
      IF(YPMAX.GE.2.0) YP=2.*TAU/TAUF
      RETURN
      END
C
      SUBROUTINE TAUOC(S,TOC)
      DIMENSION S(4)
      S1=S(1)
      S2=S(2)
      S3=S(3)
      S4=S(4)
      TOC=(SQRT(S1*S1+S2*S2+S3*S3+2.*S4*S4))/SQRT(3.)
      RETURN
      END
C
      SUBROUTINE CALYF(PA,TY,TF,PC,TAUY,TAUF,NTP)
      DIMENSION PA(1),TY(1),TF(1)
C     EMA PA,TY,TF,NTP
      K=1
100   IF(PC.GE.P(K).AND.PC.LE.P(K+1)) GO TO 200
      IF(PC.GT.P(K+1)) K=K+1
      IF(PC.LT.P(K)) K=K-1
      IF(K.LT.NTP) GO TO 100
      WRITE(6,1)
1     FORMAT(22H K IS GREATER THEN NTP)
      STOP
200   CONTINUE
      DP=P(K+1)-P(K)
      DPC=PC-P(K)
      TAUY= TY(K)+(TY(K+1)-TY(K))*DPC/DP
      TAUF= TF(K)+(TF(K+1)-TF(K))*DPC/DP
      RETURN
      END
```

```
C
      SUBROUTINE DEVST(DE,G,DS)
      DIMENSION DE(4),DS(4)
      DO 100 I=1,3
      DS(I)= 2.*G*DE(I)
100   CONTINUE
      DS(4)=G*DE(4)
      RETURN
      END
C
      SUBROUTINE ADDVT(A,B,C)
      DIMENSION A(4),B(4),C(4)
      DO 100 I=1,4
      C(I)= A(I)+B(I)
100   CONTINUE
      RETURN
      END
C
C $EMA /MDCPL/
      SUBROUTINE MTPK4(PS,DSTRN,DP,NFIRST,NFSTF,NMAT,PROP)
      COMMON/MDCPL/ CLM(11,2,5),CUM(11,2,5),TYF(11,3,5),NLP(5),
     *               NUP(5),NTP(5),NCO(5)
      DIMENSION DSTRN(4),DP(4,4),PS(1),PROP(1)
C     EMA PS,PROP
      N=NMAT
      POSNR=PROP(8)
      EQNO=PROP(9)
      C=PROP(10)
      D=PROP(11)
      SVMLL=PROP(12)
      IF(NFIRST.NE.0)GO TO 300
C
C
C     NFIRST=0,  ASSUME LOADING
C
C
      PS(7)=PS(2)
      PS7 = PS(7)
C
      CALL MDLUS(CLM(1,1,N),CLM(1,2,N),NLP(N),PS7,CSM)
C
      BK=(1.+POSNR)*CSM/(3.*(1.-POSNR))
      G=(1.-2.*POSNR)*CSM/(2.*(1.-POSNR))
C
      GOTO 800
C
C
C     NFIRST > 0
C
C
300   CONTINUE
```

```
C
      CALL UNIAX(PS(1),DSTRN,CLM(1,1,N),NLP(N),CUM(1,1,N),NUP(N),
     *           PS(7),POSNR,EQNO,C,D,SVMLL,BK,G)
C
C
C     CALCULATE STRESS-STRAIN MATRIX
C
C
800   CONTINUE
C
      IF(NFSTF.EQ.0) RETURN
      CALL ELAST(BK,G,DP)
      RETURN
      END
C
      SUBROUTINE UNIAX(STRS,DSTRN,CLM,NLP,CUM,NUP,S2MAX,POSNR,
     *                 EQNO,C,D,SVMLL,BK,G)
C
      DIMENSION STRS(4),DSTRN(4),CLM(11,2),CUM(11,2)
C     EMA STRS,CLM,NLP,CUM,NUP,S2MAX
C
C
C
C     INPUT:
C           STRS(4) :        (OLD STRESS)
C           DSTRN(4):        (CURRENT STRAIN INCREMENT)
C           CLM(NLP,2):       SV-ML CURVE
C           CUM(NUP,2):       SV-MU CURVE IF EQNO=1.
C                             SVMAX-MU CURVE IF EQNO=2.
C           FOR EQNO=3.
C           C,D,SVMLL:        CONSTANTS FOR EQUATION OF UNLOADING
C                             CONSTRAINED MODULUS
C           S2MAX:            PREVIOUS MAXIMUM SV
C           POSNR:            POISSONS RATIO (ASSUME CONSTANT)
C
C     OUTPUT=
C           STRS(4):          (UPDATE.STRESS)
C           S2MAX:            UPDATE SV MAX
C           BK:               UPDATE BULK MODULUS
C           G:                UPDATE SHEAR MODULUS
C
C
C
C     SET THE NUMBER OF STRAIN SUBINCREMENT INTO 10
C
C
      NDIV=10
      DDE2=DSTRN(2)/NDIV
C
      S2=STRS(2)
      DS2=0.0
```

```
C
C
      DO 1000 II=1,NDIV
C
      IF(S2.GE.S2MAX) GO TO 400
C
C     UNLOADING, 0.0 < S2 < S2 MAX
C
      IF(S2.LE.0.0) GO TO 200
C
      IEQNO=INT(EQNO)
      GOTO(10,20,30),IEQNO
      WRITE(6,1) IEQNO
1     FORMAT(24H UNLOADING EQUATION NO =,I5,13H IS NOT VALID)
      STOP
10    CONTINUE
C
C     UNLOADING CONSTRAINED MODULUS BASED ON CURRENT
C     VERTICAL STRESS
C
      CALL MDLUS(CUM(1,1),CUM(1,2),NUP,S2,CSM)
      GO TO 600
20    CONTINUE
C
C     UNLOADING CONSTRAINED MODULUS BASED ON PREVIOUS
C     MAXIMUM VERTICAL STRESS
C
      CALL MDLUS(CUM(1,1),CUM(1,2),NUP,S2,CSM)
      GO TO 600
30    CONTINUE
C
C     UNLOADING CONSTRAINED MODULUS BASED ON
C     EMPIRICAL EQUAITON (BY SCOTT BLOUIN)
C
      IF(S2MAX.GE.SVMLL) CSM=C*(S2MAX**D)
      IF(S2MAX.LT.SVMLL) CSM=C*(SVMLL**D)
      GO TO 600
200   CONTINUE
C
C     UNLOADING AND LIQUEFIED,     S2 <= 0.0 < S2 MAX
C
C
      CSM=0.0
       S2=0.0
      DS2=0.0
      STRS(1)=0.0
      STRS(2)=0.0
      STRS(3)=0.0
      GO TO 1000
400   CONTINUE
C
```

```
C     LOADING,     S2 >= S2 MAX.
C
C
      CALL MDLUS(CLM(1,1),CLM(1,2),NLP,S2,CSM)
      GOTO 600
C
C     UPDATE VERTICAL STRESS AND S2 MAX.
C
600   CONTINUE
      DDS2=CSM*DDE2
      DS2=DS2+DDS2
      S2=S2+DDS2
      IF(S2.GT.S2MAX) S2MAX=S2
C
C
1000  CONTINUE
C
C     UPDATE ALL STRESS
C
      STRS(1)=STRS(1)+DS2*POSNR/(1.-POSNR)
      STRS(2)=S2
      STRS(3)=STRS(1)
      STRS(4)=0.0
C
C     UPDATE BULK AND SHEAR MODULI
C
      BK=(1.+POSNR)*CSM/(3.*(1.-POSNR))
      G=(1.-2.*POSNR)*CSM/(2.*(1.-POSNR))
C
      RETURN
      END
C $EMA /TENCT/
      SUBROUTINE CUTEN(NMPK,D,ST,IP,STIFAC,SHEFAC,PMN,FG)
C
C     COMPUTE PRINCIPAL STRESS AND ITS ANGLE
C     CHECK TENSION, IF THERE IS TENSION, CUTOFF AND MODIFY D-MATRIX
C
      COMMON /TENCT/ T(4,4),H(4,4)
      DIMENSION D(4,4),ST(4),S(4)
C     EMA ST,FG
      CALL DUPVX(S,ST)
      IF(NMPK.EQ.1) CALL SIGN(S,4)
C     COMPUTE PRINCIPAL STRESSES AND ITS ANGLE
      P1 = S(1)+S(2)
      P2 = S(1)-S(2)
      IF(P2.EQ.0.0) GO TO 20
      ANGX = 0.5*ATAN(2.*S(4)/P2)
      GO TO 40
   20 ANGX = 0.7854
   40 PX = 0.5*P1+0.5*P2*COS(2.*ANGX)+S(4)*SIN(2.*ANGX)
      ANGY = ANGX+1.5708
      PY = 0.5*P1+0.5*P2*COS(2.*ANGY)+S(4)*SIN(2.*ANGY)
      PZ = S(3)
C     CHECK TENSION, IF THERE IS TENSION, CUTOFF
      IF(PX.LT.PMN.AND.PY.LT.PMN) GO TO 470
      IF(FG.NE.-1000.) GO TO 60
      IF(PY.GT.PX) GO TO 50
      FG = ANGX
      GO TO 60
   50 FG = ANGY
   60 CONTINUE
      IF(PX.GT.PMN) PX = PMN
      IF(PY.GT.PMN) PY = PMN
      IF(PZ.GT.PMN) PZ = PMN
C     BACK-TRANSFER STRESSES FROM PRINCIPAL TO X-Y COORDINATE
      C = COS(ANGX)
      S = SIN(ANGX)
      C2 = C*C
      S2 = S*S
      CS = C*S
      S(1) = C2*PX+S2*PY
      S(2) = S2*PX+C2*PY
      S(3) = PZ
      S(4) = CS*(PX-PY)
      IF(IP.NE.4) GO TO 450
      S(3) = S(2)
      S(4) = 0.0
  450 CONTINUE
      IF(NMPK.EQ.1) CALL SIGN(S,4)
      CALL VXDUP(ST,S)
C     TRANSFER D-MATRIX IN PRINCIPAL OR CRACKED COORDINATE TO X-Y COORDINA
  470 IF(FG.EQ.-1000.) RETURN
      D(1,1) = D(1,1)/STIFAC
      D(1,2) = D(1,2)/STIFAC
      D(1,3) = D(1,3)/STIFAC
      D(4,4) = D(4,4)/SHEFAC
      C = COS(FG)
      S = SIN(FG)
      C2 = C*C
      S2 = S*S
      CS = C*S
      DO 48C I=1,4
      DO 480 J=I,4
  480 D(J,I) = D(I,J)
C
      T(1,1) = C2
      T(1,2) = S2
      T(1,3) = 0.0
      T(1,4) = CS
      T(2,1) = S2
      T(2,2) = C2
      T(2,3) = 0.0
      T(2,4) = -CS
      T(3,1) = 0.0
      T(3,2) = 0.0
      T(3,3) = 1.0
      T(3,4) = 0.0
      T(4,1) = -2.*CS
      T(4,2) = 2.*CS
      T(4,3) = 0.0
      T(4,4) = C2-S2
      DO 500 J=1,4
      T1=D(1,1)*T(1,J)+D(1,2)*T(2,J)+D(1,3)*T(3,J)+D(1,4)*T(4,J)
      T2=D(2,1)*T(1,J)+D(2,2)*T(2,J)+D(2,3)*T(3,J)+D(2,4)*T(4,J)
      T3=D(3,1)*T(1,J)+D(3,2)*T(2,J)+D(3,3)*T(3,J)+D(3,4)*T(4,J)
      T4=D(4,1)*T(1,J)+D(4,2)*T(2,J)+D(4,3)*T(3,J)+D(4,4)*T(4,J)
      DO 500 I=J,4
      H(I,J) = T(1,I)*T1+T(2,I)*T2+T(3,I)*T3+T(4,I)*T4
  500 H(J,I) = H(I,J)
C
      DO 600 I=1,4
      DO 600 J=1,4
  600 D(I,J) = H(I,J)
C
      RETURN
      END
```

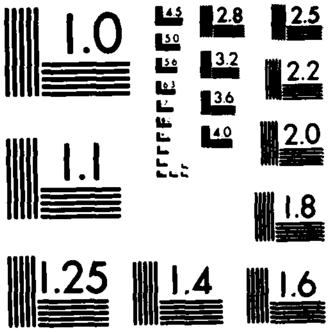
```
C FTN7X,L
      SUBROUTINE MTPK5(PS,DSTRN,DP,NFIRST,NFSTF,BKE,GE,PROP,IP)
      COMMON/MATPR/EMP(7),A2,B2,C2
      DIMENSION DSTRN(4),DP(4,4),PS(1),PROP(1),ICH(4),C(4,4),
     *          EPS(4),DELEPS(4),SIG(4)
C     EMA PS,PROP
      DATA ICH/1,2,4,3/
C
C
C     STRESS AND STRAIN CONDITIONS:
C            PLANE STRESS:        IP=1, ITYP2D=2
C            PLAIN STRAIN:        IP=2, ITYP2D=1
C            AXIAL SYMMETRY:      IP=3, ITYP2D=0
C            SPHERICAL SYMMETRY: IP=4, ITYP2D=0
C
C
      IF(IP.EQ.1) ITYP2D=2
      IF(IP.EQ.2) ITYP2D=1
      IF(IP.EQ.3) ITYP2D=0
      IF(IP.EQ.4) ITYP2D=0
C
C
C     CALCULATE V AND E FROM BK AND G
C
C
      V=(3.*BKE-2.*GE)/(6.*BKE+2.*GE)
      E= 3.*BKE*(1.-2.*V)
      EMP(1)=E
      EMP(2)=V
C
C     ASSIGN   N,ALPA,B,KAPA,AND K
C
C
      DO 20 I=3,7
      II=I+5
      EMP(I)=PROP(II)
20    CONTINUE
C
C
C     CHANGE STRESSSES, STRAINS, AND ELASTIC-PLASTIC FLAG
C     FROM TPDAP TO ARA2D
C
C
      DO 100 I=1,4
      II= ICH(I)
      SIG(I)= -PS(II)
      DELEPS(I)= -DSTRN(II)
100   CONTINUE
C
      IF(NFIRST.EQ.0) PS(7) = 1.0
      PS7 = PS(7)
      IPEL= INT(PS7)
C
      CALL SZERO(EPS,4)
C
C     CALL ARA2D MODEL
C
      CALL ARA2D(IPEL,EPS,DELEPS,SIG,ITYP2D,C)
C
C     CHANGE STRESS-STRAIN MATRIX FROM ARA2D TO TPDAP
C
      DO 200 I=1,4
      II= ICH(I)
      DO 200 J=1,4
      JJ= ICH(J)
      DP(I,J)=C(II,JJ)
200   CONTINUE
C
C
C     CHANGE STRESSES AND ELASTO-PLASTIC FLAG FROM ARA2D TO TPDAP
C
C
      DO 300 I=1,4
      II= ICH(I)
      PS(I)= -SIG(II)
300   CONTINUE
C
      PS(7)=FLOAT(IPEL)
C
      RETURN
      END
      SUBROUTINE ARA2D (IPEL,EPS,DELEPS,SIG,ITYP2D,C)
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .      SIG     STRESSES AT THE END OF THE PREVIOUS UPDATE           .
C .      EPS     STRAINS  AT THE END OF THE PREVIOUS UPDATE           .
C .      RATIO   PART OF STRAIN INCREMENT TAKEN ELASTICALLY           .
C .      DELEPS  INCREMENT IN STRAINS                                 .
C .      DELSIG  INCREMENT IN STRESSES, ASSUMING ELASTIC BEHAVIOR     .
C .                                                                   .
C .      PROP(1) YOUNG S  MODULUS                                     .
C .      PROP(2) POISSON S  RATIO                                     .
C .      PROP(3) H1 (N)                                               .
C .      PROP(4) H2 (ALPHA)                                           .
C        PROP(5) H3 (BETA)
C        PROP(6) H4 (KAPA)
C        PROP(7) H5 ( K   )
C .                                                                   .
C .      IPEL    =  1, MATERIAL ELASTIC,
C .               =  2, MATERIAL PLASTIC
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C
      COMMON /MATPR/ PROP(7),A2,B2,C2
C
      DIMENSION SIG(4),EPS(4),C(4,4)
      DIMENSION TAU(4),DELSIG(4),DELEPS(4),DEPS(4)
C
      CALL CSIGN(SIG)
      CALL CSIGN(EPS)
      CALL CSIGN(DELEPS)
C
      IF(PROP(3).NE.0.0) CALL CUTNX(SIG)
C
      YM=PROP(1)
      PV=PROP(2)
      G =YM/(1.+PV)/2.
      BM=YM/(1.-2.*PV)/3.
      A2=BM+4.*G/3.
      B2=BM-2.*G/3.
      C2=G
C
C        1. UPDATE STRAINS
C
      DO 120 I=1,4
  120 EPS(I) = EPS(I) +  DELEPS(I)
C
C
C        2. CALCULATE THE STRESS INCREMENT,
C           ASSUMING ELASTIC BEHAVIOR
C
      DELSIG(1) = A2*DELEPS(1) + B2*DELEPS(2)
      DELSIG(2) = B2*DELEPS(1) + A2*DELEPS(2)
      DELSIG(3) = C2*DELEPS(3)
      DELSIG(4) = B2 * (DELEPS(1)+DELEPS(2))
      IF (ITYP2D.EQ.1) GO TO 150
      DELSIG(1) = DELSIG(1) + B2*DELEPS(4)
      DELSIG(2) = DELSIG(2) + B2*DELEPS(4)
      DELSIG(4) = DELSIG(4) + A2*DELEPS(4)
C
C
C        3. CALCULATE TOTAL STRESSES,
C           ASSUMING ELASTIC BEHAVIOR
C
  150 CONTINUE
      DO 160 I=1,4
  160 TAU(I) = SIG(I) + DELSIG(I)
C
C        4. CHECK WHETHER *TAU* STATE OF STRESS FALLS
```

```
C           OUTSIDE THE LOADING SURFACE
C
C     IF(IPEL.EQ.2) GO TO 165
C
      CALL FORMF(TAU,FT)
C
      IF(IPEL.EQ.1.AND.FT.LE.0.0) GO TO 170
      IF(IPEL.EQ.1.AND.FT.GT.0.0) GO TO 300
  165 RATIO = 0.0
      GO TO 320
C
C     STATE OF STRESS WITHIN LOADING SURFACE  -  ELASTIC BEHAVIOR
C
  170 IPEL=1
      DO 180 I=1,4
  180 SIG(I) = TAU(I)
      GO TO 400
C
C
C     STATE OF STRESS OUTSIDE LOADING SURFACE  -  PLASTIC BEHAVIOR
C
C
C     DETERMINE PART OF STRAIN TAKEN ELASTICLY
C
  300 IPEL=2
C
      CALL SHOOT(SIG,TAU,RATIO)
C
  320 CONTINUE
      DO 350 I=1,4
  350 TAU (I) = SIG(I) + RATIO*DELSIG(I)
C
C     *TAU* NOW CONTAINS (PREVIOUS STRESSES +
C     STRESSES DUE TO ELASTIC STRAIN INCREMENTS)
C
C
C        5. CALCULATE PLASTIC STRESSES
C
C        DETERMINE INCREMENT INTERVAL
C
  370 CONTINUE
      M = 10
      XM = (1.-RATIO)/FLOAT(M)
C
C
      DO 380 I=1,4
  380 DEPS(I) = XM*DELEPS(I)
      IF(ITYP2D.EQ.1) DEPS(4) = 0.0
C
C
C     .....  CALCULATION OF ELASTOPLASTIC STRESSES   .....   (START)
```

```
C
      DO 600 IM=1,M
C
      CALL DMTRX (TAU,DEPS,C)
C
      DO 560 I=1,4
      DO 560 J=1,4
  560 TAU(I) = TAU(I) + C(I,J) * DEPS(J)
C
C
C     CORRECTION
C
      CALL DRFTA(TAU)
C
C
  600 CONTINUE
C
C     .....  CALCULATION OF ELASTOPLASTIC STRESSES  .....  ( END )
C
      DO 390 I=1,4
  390 SIG(I) = TAU(I)
C
C
C     CHECK TENSILE STRESSES
C
  400 CONTINUE
      IF(PROP(3).NE.0.0) CALL CUTNX(SIG)
C
      CALL CSIGN(SIG)
      CALL CSIGN(EPS)
C
C        7. FORM THE MATERIAL LAW
C
      IF (IPEL.EQ.1) GO TO 450
C
C     ELASTO-PLASTIC
C
      CALL CSIGN(SIG)
C
      CALL DMTRX (SIG,DEPS,C)
C
      CALL CSIGN(SIG)
C
      RETURN
C
C     ELASTIC
C
  450 CONTINUE
      CALL SZERO(C,16)
C
      C(1,1)=A2
      C(2,1)=B2
      C(1,2)=B2
      C(2,2)=A2
      C(3,3)=C2
      C(1,4)=B2
      C(2,4)=B2
      C(4,1)=B2
      C(4,2)=B2
      C(4,4)=A2
C
      RETURN
      END
      SUBROUTINE CUTNX(S)
      DIMENSION S(1)
C
      S(1) = -S(1)
      S(2) = -S(2)
      S(3) = -S(3)
C
C
C     COMPUTE PRINCIPAL STRESSES AND ANGLES
C
      P1 = S(1)+S(2)
      P2 = S(1)-S(2)
      IF(ABS(P2).LT.1.E-8) GO TO 20
      ANGX = 0.5*ATAN(2.*S(3)/P2)
      GO TO 40
   20 ANGX = 0.7854
   40 CONTINUE
      PX = 0.5*P1+0.5*P2*COS(2.*ANGX)+S(3)*SIN(2.*ANGX)
      ANGY = ANGX+1.5708
      PY = 0.5*P1+0.5*P2*COS(2.*ANGY)+S(3)*SIN(2.*ANGY)
C
C     CHECK TENSION, IF THERE IS TENSION, CUT OFF
C
      IF(PX.GT.0.0) PX = 0.0
      IF(PY.GT.0.0) PY = 0.0
C
C     BACK-TRANSFER STRESSES FROM PRINCIPAL TO X-Y COORDINATE
C
      C = COS(ANGX)
      S = SIN(ANGX)
      C2 = C*C
      S2 = S*S
      CS = C*S
C
      S(1) = C2*PX+S2*PY
      S(2) = S2*PX+C2*PY
      S(3) = CS*(PX-PY)
C
      S(1) = -S(1)
      S(2) = -S(2)
      S(3) = -S(3)
C
      RETURN
      END
      SUBROUTINE PSTRN(E,EMAX,EMIN)
      DIMENSION E(1)
C
      E1 = E(1)+E(2)
      E2 = E(1)-E(2)
      E3 = 0.5*E(3)
      IF(ABS(E2).LT.1.E-8) GO TO 20
      ANGX = 0.5*ATAN(2.*E3/E2)
      GO TO 40
   20 ANGX = 0.7854
   40 CONTINUE
      EX = 0.5*E1+0.5*E2*COS(2.*ANGX)+E3*SIN(2.*ANGX)
      ANGY = ANGX+1.5708
      EY = 0.5*E1+0.5*E2*COS(2.*ANGY)+E3*SIN(2.*ANGY)
      IF(EX.GT.EY) GO TO 60
C     EMAX = EY
      EMIN = EX
      GO TO 80
   60 EMAX = EX
      EMIN = EY
   80 CONTINUE
      RETURN
      END
      SUBROUTINE DMTRX (TAU,DEPS,DP)
C
C        FORMS THE ELASTO-PLASTIC MATERIAL MATRIX
C
      COMMON /MATPR/ PAR(7),A2,B2,C2
C
      DIMENSION TAU(4),DEPS(4),DP(4,4),A(4),DP1(4),DA1(4)
C
      CALL SZERO(DP,16)
C
      DP(1,1) = A2
      DP(2,1) = B2
      DP(1,2) = B2
      DP(2,2) = A2
      DP(3,3) = C2
      DP(1,4) = B2
      DP(4,1) = B2
      DP(2,4) = B2
      DP(4,2) = B2
      DP(4,4) = A2
C
      CALL VECTA(TAU,A)
C
```

AD-A174 749 EXPERIMENTAL AND THEORETICAL RESPONSE OF MULTIPHASE POROUS MEDIA TO DYNAM (U) APPLIED RESEARCH ASSOCIATES INC SOUTH ROYALTON VT NEW ENGLAND K J KIM ET AL 4/4

UNCLASSIFIED 13 AUG 86 ARA-5967-86 AFOSR-TR-86-0665 F/G 8/13 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

```
      CALL MLPLY(DP,DEPS,DP1,4)
C
      CALL DOT(A,DP1,DE1,4)
C
      CALL MLPLY(DP,A,DA1,4)
C
      CALL DOT(A,DA1,X1,4)
C
      DLMDA = DE1/X1
C
C     IF(DLMDA.LE.0.0) GO TO 250
C
      DO 200 I=1,4
      DO 200 J=1,4
  200 DP(I,J) = DP(I,J)-DA1(I)*DA1(J)/X1
C
C
  250 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE SHOOT(STRSL,STRSR,RATIO)
      DIMENSION STRSL(4),STRSR(4),STRSX(4)
C
C
C
      NN = 10.
      XL = 0.0
      XR = 1.0
C
      CALL FROOT(XL,XR,STRSL,STRSR,STRSX,RATIO,NN)
C
C
      RETURN
      END
      SUBROUTINE DRFTA(STRESS)
      DIMENSION STRESS(4),STRSL(4),STRSR(4),STRSX(4)
C
C
      NN = 5
      DX = 0.1
      CALL DCOPY(STRSX,STRESS)
      XX = 1.0
      CALL FORMF(STRSX,FX)
C
      IF(FX.LT.0.0) GO TO 500
      XR = XX
      CALL DCOPY(STRSR,STRSX)
      K = 0
  100 XX = XX-DX
```

```
      K = K+1
      IF(K.GT.100) GO TO 900
      CALL FCOPY(STRSX,STRESS,XX)
      CALL FORMF(STRSX,FX)
      IF(FX.GE.0.0) GO TO 100
      XL = XX
      CALL DCOPY(STRSL,STRSX)
      GO TO 800
C
  500 CONTINUE
      XL = XX
      CALL DCOPY(STRSL,STRESS)
      K = 0
  700 XX = XX+DX
      K = K+1
      IF(K.GT.100) GO TO 900
      CALL FCOPY(STRSX,STRESS,XX)
      CALL FORMF(STRSX,FX)
      IF(FX.LE.0.0) GO TO 700
      XR = XX
      CALL DCOPY(STRSR,STRSX)
C
  800 CALL FROOT(XL,XR,STRSL,STRSR,STRESS,R,NN)
C
      RETURN
  900 WRITE(6,*) " WRONG DRIFT CORRECTION IN DRFT-A ",XR,XX,FX,K
      STOP
      END
C
      SUBROUTINE DCOPY(A,B)
      DIMENSION A(4),B(4)
      DO 100 I=1,4
  100 A(I) = B(I)
      RETURN
      END
C
      SUBROUTINE ADDST(A,B,R,C)
      DIMENSION A(4),B(4),C(4)
      DO 100 I=1,4
  100 A(I) = B(I)+R*C(I)
      RETURN
      END
      SUBROUTINE FORMF(S,F)
      COMMON /MATPR/ PAR(2),A1,A2,A3,A4,X
      DIMENSION S(4)
C
      CALL INVRS(S,P,Q,RT,T)
C
C     F = Q-((A2+A3*P)**A1+A4)*RT
      IF(A1.EQ.0.0) AP = 1.0
      IF(A1.EQ.0.5) AP = SQRT(A2+A3*P)
      IF(A1.EQ.1.0) AP = A2+A3*P
      F = Q-(AP+A4)*RT
C
C
      RETURN
      END
C
      SUBROUTINE INVRS(S,P,Q,RT,T)
      COMMON /MATPR/ PAR(2),A1,A2,A3,A4,X
      DIMENSION S(4)
C
      P = (S(1)+S(2)+S(4))/3.
      S12 = S(1)-S(2)
      S23 = S(2)-S(4)
      S31 = S(4)-S(1)
C
      Q = SQRT(.5*(S12*S12+S23*S23+S31*S31)+3.*S(3)*S(3))
C
      IF(Q.GT.1.0E-15) GO TO 10
      T = 0.0
      RT = 1.0
      RETURN
   10 CONTINUE
C
      SX = S(1)-P
      SY = S(2)-P
      SZ = S(4)-P
      Z3 = SX*SY*SZ-SZ*S(3)*S(3)
      ST3 = -13.5*Z3/(Q*Q*Q)
      IF(ST3.GT.1.0) ST3 = 1.0
      IF(ST3.LT.-1.0) ST3 = -1.0
C
      T = (ASIN(ST3))/3.0
C
      A = 1.0-X*X
      B = 2.0*X-1.0
      C = 5.0*X*X-4.0*X
      SQ3 = SQRT(3.)
      CS = SQ3*COS(T)+SIN(T)
      TS = 2.+COS(2.*T)+SQ3*SIN(2.*T)
      BS = SQRT(A*TS+C)
      RN = A*CS+B*BS
      RD = A*TS+B*B
C
      RT = RN/RD
C
      RETURN
      END
C
      SUBROUTINE VECTA(S,A)
      COMMON /MATPR/ PAR(2),A1,A2,A3,A4,X
      DIMENSION S(4),A(4)
C
C     DERIVATIVES OF YIELD FUNCTION W.R.T. STRESS
C
      CALL INVRS(S,P,Q,RT,T)
C
      C1 = COS(T)
      C2 = COS(2.*T)
      C3 = COS(3.*T)
      S1 = SIN(T)
      S2 = SIN(2.*T)
      SQ3 = SQRT(3.)
C
C     FP = -A1*A3*RT/((A2+A3*P)**(1.-A1))
      IF(A1.EQ.0.0) FP = 0.0
      IF(A1.EQ.0.5) FP = -0.5*A3*RT/SQRT(A2+A3*P)
      IF(A1.EQ.1.0) FP = -A3*RT
C
      IF(Q.GE.1.0E-15) GO TO 100
      A(1) = FP/3.
      A(2) = FP/3.
      A(3) = 0.0
      A(4) = FP/3.
      RETURN
C
  100 CONTINUE
C
      FQ = 1.0
C
      AA = 1.0-X*X
      BB = 2.*X-1.0
      CC = 5.*X*X-4.*X
      TS = 2.+C2+SQ3*S2
      BS = SQRT(AA*TS+CC)
      RD = AA*TS+BB*BB
      CSC = SQ3*C2-S2
      RNT = AA*(C1-SQ3*S1)+AA*BB*CSC/BS
      RDT = 2.*AA*CSC
      RWT = (RNT-RT*RDT)/RD
C     FT = -((A2+A3*P)**A1+A4)*RWT
      IF(A1.EQ.0.0) AP = 1.0
      IF(A1.EQ.0.5) AP = SQRT(A2+A3*P)
      IF(A1.EQ.1.0) AP = A2+A3*P
      FT = -(AP+A4)*RWT
C
      Q2 = Q*Q
      Q3 = Q*Q2
      Q4 = Q*Q3
      SX = S(1)-P
      SY = S(2)-P
      SZ = S(4)-P
      SXY = SX*SY
      SXZ = SX*SZ
      SYZ = SY*SZ
      TXY = S(3)*S(3)
C
      Z3 = SX*SY*SZ-SZ*S(3)*S(3)
C
      ZX = SY*SZ+Q2/9.
      ZY = SX*SZ+Q2/9.
      ZXY = -2.*SZ*S(3)
      ZZ = SX*SY-TXY+Q2/9.
C
      F1 = FP/3.
      F2 = FQ+13.5*Z3*FT/(Q4*C3)
      FN2 = 1.5*F2/Q
      F3 = -4.5*FT/(Q3*C3)
C
      A(1) = F1+FN2*SX+F3*ZX
      A(2) = F1+FN2*SY+F3*ZY
      A(3) = 2.*FN2*S(3)+F3*ZXY
      A(4) = F1+FN2*SZ+F3*ZZ
C
      RETURN
      END
C
      SUBROUTINE MLPLY(C,A,B,N)
      DIMENSION C(N,N),A(N),B(N)
      DO 100 I=1,N
      B(I) = 0.0
      DO 100 J=1,N
  100 B(I) = B(I)+C(I,J)*A(J)
      RETURN
      END
C
      SUBROUTINE DOT(X,Y,FF,N)
      DIMENSION X(N),Y(N)
      FF = 0.0
      DO 100 I=1,N
  100 FF = FF+X(I)*Y(I)
      RETURN
      END
C
C
      SUBROUTINE SZERO(A,N)
      DIMENSION A(N)
      DO 100 I=1,N
  100 A(I) = 0.0
      RETURN
      END
C
      SUBROUTINE FCOPY(A,B,FAC)
      DIMENSION A(4),B(4)
C
      DO 100 I=1,4
  100 A(I) = FAC*B(I)
C
      RETURN
      END
C
C
      SUBROUTINE FROOT(XXL,XXR,STRSL,STRSR,STRSX,R,NN)
      DIMENSION STRSL(4),STRSR(4),STRSX(4)
C
      XL = XXL
      XR = XXR
C
      DO 100 I=1,NN
C
      XX = XL+(XR-XL)*0.5
      DO 50 J=1,4
   50 STRSX(J) = STRSL(J)+(XX-XXL)*(STRSR(J)-STRSL(J))/(XXR-XXL)
      CALL FORMF(STRSX,FX)
C
      IF(FX.GT.0.0) GO TO 60
      XL = XX
      GO TO 100
   60 CONTINUE
      XR = XX
C
  100 CONTINUE
C
C
      R = XX
C
      RETURN
      END
C
      SUBROUTINE CSIGN(A)
      DIMENSION A(4)
      DO 100 I=1,4
  100 A(I) = -A(I)
      RETURN
      END
```

REFERENCES

Blouin, S.E., and K.J. Kim, "Undrained Compressibility of Saturated Soil," Draft Report to Defense Nuclear Agency, Washington, DC, 13 February, 1984.

Bridgman, P.W., The Physics of High Pressure, Dover Publications, Inc., New York, NY, 1931.

Simmons, G., and H. Wang, Single Crystal Elastic Constants and Calculated Aggregate Properties: A Handbook, Second Edition, The M.I.T. Press, Cambridge, Massachusetts and London, England, 1971.

Blouin, S.E., et.al., "Laboratory Investigation of the Mechanical Properties of Enewetak Sand," Draft Report to Air Force Office of Scientific Research, June 1984.

Blouin, S.E., and J.D. Shinn II, "Explosion-Induced Liquefaction," Report to Air Force Office of Scientific Research, Washington, DC, 1983.

Biot, M.A., "Theory of Propagation of Elastic Waves in Fluid Saturated Porous Solid, I, II", Journal of Acoustical Society of America, Vol. 28, pp. 168-191, 1956.

Biot, M.A., "Mechanics of Deformation and Acoustic Propagation in Porous Media," Journal of Applied Physics, Vol. 33, pp. 1482-1498, 1962a.

Biot, M.A., "Generalized Theory of Acoustic Propagation in Porous Dissipative Media", Journal of Acoustical Society of America, Vol. 34, pp. 1254-1264, 1962b.

Ward, J., "Turbulent Flow in Porous Media", in ASCE Journal of the Hydraulic Division, American Society of Civil Engineers, HY5, pp. 1-12, 1964.

Britt, J.R., "Behavior of Water in Low Pressure Region", unpublished manuscript, 1985.

Wilson, W., and D. Bradley, "Specific Volume, Thermal Expansion, and Isothermal Compressibility of Sea Water", U.S. Naval Ordinance Laboratory Report, NOLTR, pp. 66-103, 1966.

Grindley, T., and J.E. Lind, Jr., "PVT Properties of Water and Mercury", Journal of Chemical Physics, Vol. 54, pp. 3983-3989, 1971.

Bridgman, P.W., "Thermodynamic Properties of Liquid Water to 80° and 12,000 hgm", Proceedings of the American Academy of Arts and Sciences, Vol. 48, pp. 309-362, 1912.

Rice, M.H., and J.M. Walsh, "Equation of State of Water to 250 Kbars", Journal of Chemical Physics, Vol. 26, pp. 824-830, 1957.

REFERENCES (continued)

Skidmore, I.C., and E. Morris, "Experimental Equation of State Data for Uranium and Its Interpretation in the Critical Region", Thermodynamics of Nuclear Materials, International Atomic Energy Agency, Vienna, Austria, 1962.

Adams, L.H., "Equilibrium in Binary Solutions Under Pressure", American Chemistry Society Journal, Vol. 53, pp. 3769-3813, 1931.

Wood, A.B., A Textbook of Sound, G. Bell and Sons, London, England, 1930.

Kim, K.J., D.D. Piepenburg, and D.H. Merkle, "Influence of the Intermediate Principal Stress on Rock Tunnel Behavior", Draft Report to Defense Nuclear Agency, Washington, DC, November 1985.

Hoek, E. and E.T. Brown, Underground Excavations in Rock, The Institution of Mining and Metallurgy, London, England, 1982.

Kim, K.J., M.D. Davister, M.A. Rochefort, and D.D. Piepenburg, "Numerical Analysis of Nonlinear Linear-Medium Interaction, Part II: Cylindrical Tunnel Subjected to Axisymmetric Loading" (Verification and Parameter Study), Draft Report to Defense Nuclear Agency, Washington, DC, July 1986.

Rischbieter, F., et.al., "Studies of Soil Liquefaction by Shock Wave Loading", Fifth International Symposium on Military Applications of Blast Simulation, Vol. III, Royal Swedish Fortifications Administration, Stockholm, Sweden, May, 1977.

Kim, K.J., "Finite Element Analysis of Nonlinear Consolidation", PhD thesis, University of Illinois, Urbana-Champaign, 1982.

END

1-87

DTIC